IL MENSILE JACKSON PER GLI UTENTI DI AMIGA

- TUTTE LE NOVITA' DELL'AMIGA '92 DI BERLINO
- INTRODUZIONE ALLO STANDARD MIDI
- IN PROVA: REAL 3D • ANIMAKER • AMIGA 600 • DCTV PAL • VIDEO DIRECTOR
- *TransAction* LE PAGINE DEL PROGRAMMATORE
- ON DISK:
  - PC TASK - FANTASTICO EMULATORE PC
  - SPACE WAR - SFIDA ALL'ULTIMA ASTRONAVE
  - VIEW 3.0 - VISUALIZZA OGNI TIPO DI IMMAGINE
  - MAGIC REQUESTER - NUOVO REQUESTER PER AMIGA
  - TURBODISK - VELOCIZZA IL DISK DRIVE
  - E... ALTRI FANTASTICI PROGRAMMI

RIVISTA UFFICIALMENTE RICONOSCIUTA DA COMMODORE ITALIANA

SPEDIZIONE IN ABBONAMENTO POSTALE GRUPPO III/70 TAXE PERCUE (TASSA RISCOSSA) MILANO CMP ROSERIO

EDITORIALE

AMIGA 600? SI' GRAZIE!

Proprio pochi giorni prima di andare in stampa, è arrivato in redazione (di venerdì pomeriggio) l'ultimo nato in casa Commodore: Amiga 600HD.

La mia prima impressione è stata di stupore: la confezione è molto piccola e leggera.

Naturalmente, come un bimbo con il giocattolo nuovo, mi sono "tuffato" a collegare il neonato; seconda sorpresa: l'hard disk interno è di una comodità eccezionale.

Ma si sa, di sabato la redazione chiude: l'unico modo per continuare a "giocare" con il 600 era portarsi a casa il "gioiellino" e così ho fatto.

Fuori dagli uffici, il 600 dimostra tutta la sua flessibilità: si può collegare a qualsiasi TV e la resa è quasi come quella di un monitor dedicato. L'hard disk interno è così utile, che se ne dimentica la lentezza.

E' talmente poco ingombrante che l'ho portato a casa di amici e, invece di uscire, ci siamo fermati a giocare...

Ora, mentre scrivo questo editoriale, il giocattolino sta comodamente in un angolino della mia scrivania, senza occupare eccessivo spazio. Peccato che devo restituirlo, ma, probabilmente, il mio prossimo acquisto sarà proprio un Amiga 600: il mio vecchio 500 ha trovato un degno compagno e rivale.

Per finire qualche errata corrige: sul numero 33, pagina 69, si sostiene erroneamente che il prezzo del PC 286 della GVP è inferiore a quello dell'ATOnce Plus. Ci scusiamo con i lettori, ma il prezzo presunto, che ci era stato comunicato inizialmente, era molto inferiore a quello del listino finale. A pagina 67 compaiono due errori nel riquadro del Diskpeed: alla quarta riga il valore 3 era invece 43; alla undicesima riga, quinta colonna, il valore esatto è 728784 e non 72878. Ci scusiamo, infine, per l'errore di stampa della foto apparsa a pagina 66 del numero 32.

Massimiliano Anticoli

Direttore Responsabile: Paolo Reina
Coordinamento Tecnico e Redazionale: Massimiliano Anticoli
Tel. 02/6948.260
Redazione: Romano Tenca (TransAction) - Simone Crosignani
Segreteria di redazione e coordinamento estero:
Loredana Ripamonti - Tel. 02/6948.254
Art Director: Silvana Corbelli
Grafica, copertina, impaginazione elettronica:
Alessandro Fiore
Collaboratori: Antonello Biancalana, Paolo Canali, Daniele Cassanelli (Inserto), Alberto Geneletti, Antonello Jannone, Aldo e Andrea Laus, Diego Montefusco, Stefano Paganini, Domenico Pavone, Gabriele Ponte, Marco Pugliese, Paul Rigby, Stefano Riva, Nicola Salmoria, Carlo Santagostino (On Disk), Paolo Sommaruga, Gabriele Stecchi, Sebastiano Vigna, Andrew Walrond, Marco Zandonadi, Silvio Umberto Zanzi
Corrispondente dagli U.S.A.: Marshal M. Rosenthal
British Correspondent: Derek Dela Fuente

DIVISIONE PERIODICI

Presidente e Amministratore Delegato: Paolo Reina
Amministratore Delegato: Peter P. Tordoir
Group Publisher: Pierantonio Palerma
Publisher Area Consumer: Filippo Canavese
Coordinamento Operativo: Antonio Parmendola
Coordinamento Grafico: Marco Passoni
Pubblicità: Ambrogio Isacchi - Tel. 02/6948.218
Direzione Marketing e Promotion: Filippo Canavese
SEDE LEGALE
Via Rosellini, 12 - 20124 Milano
DIREZIONE - REDAZIONE
Via Pola, 9 - 20124 Milano - Tel. 02/69481
Fax: 02/6948238 Telex 316213 REINA I
PUBBLICITA'
Via Pola, 9 - 20124 Milano - Tel.: 02/6948254
ROMA - LAZIO E CENTRO SUD
Via Lago di Tana, 16 - 00199 Roma
Tel.: 06/8380547 Fax: 06/8380637
EMILIA ROMAGNA
Giuseppe Pintor Via della Chiesa, 1 - 40060 Toscanella (BO) - Tel.: 051/387790 - Fax: 051/310875
TOSCANA
Camilla Parenti - Publindustria - Via S. Antonio, 22 - 50125 Pisa - Tel.: 050/47441-49451-48194 - Fax 050/48194
INTERNATIONAL MARKETING
Stefania Scroglieri - Tel. 02/6948.229

UFFICIO ABBONAMENTI
Via Amendola, 39 - 20037 Paderno Dugnano (MI)
Fax: 02/99042386 - Tel.: 02/99043127-133 (hot line per informazioni sull'abbonamento) e 02/99044204 (sottoscrizione-rinnovo).
Tutti i giorni e venerdì dalle 9.00 alle 16.00.

Prezzo della rivista: L. 14.000 prezzo arretrato L. 28.000
Non saranno evase richieste di numeri arretrati antecedenti due anni dal numero in corso.
Abbonamento annuo Italia L. 123.200 Estero L. 246.400
I versamenti vanno indirizzati a
Gruppo Editoriale Jackson
Via Rosellini, 12 - 20124 Milano, mediante l'emissione di assegno bancario o per contanti. L'abbonamento può essere sottoscritto anche utilizzando il c/c postale 18893206

Stampa: F.B.M. (Gorgonzola)
Fotolito: Foligraph (Milano)
Distribuzione: Sodip - Via Zuretti, 25 - 20125 Milano

Il Gruppo Editoriale Jackson è iscritto al Registro Nazionale della stampa al N. 117 Vol. 2 foglio 129 in data 17/8/1982.
Spedizione in abbonamento postale gruppo III/70
Aut. Trib. di Milano n. 102 del 22/2/1988

Mensile associato all'USPI Unione Stampa Periodica Italiana

Consorzio Stampa Specializzata Tecnica

Testata aderente al C.S.S.T. non soggetta a certificazione obbligatoria per la presenza pubblicitaria inferiore al 10%

POWER COMPUTING

GVP Serie II
La nuova generazione di SCSI & RAM controllers per AMIGA 2000

Pienamente SCSI compatibile, fino a 8MB di RAM su scheda nuovo controller ad alta velocità "FAASTROM".

52MB	Quantum	954.000
105	Quantum	1.591.000
Modulo RAM da 2MB		220.000

GVP Serie II
HD Espandibile fino a 8MB RAM per A500

52MB	Quantum	1.306.000
105MB	Quantum	1.690.000
Modulo RAM da 2MB		220.000

GVP Serie II
Espansione RAM da 2 a 8MB per AMIGA 2000

2MB	400.000
4MB	620.000
6MB	840.000
8MB	1.060.000

GVP 68030
Schede acceleratrici per AMIGA 2000

68030, Coprocessore matematico 68882, controller per hard-disk SCSI o AT, Espandibile a 13,16 o 32MB 32bit RAM.

22Mhz	RAM 1MB	1.926.000
33Mhz	RAM4MB	3.684.000
Modulo RAM da 4MB 60ns 32 bit		700.000

GVP IMPACT VISION
Scheda grafica 24bit

Per A3000 e 2000, scheda grafica 16.000.000 di colori, Frame buffer 24bit 1.5MB + genlock +frame-grabber + flker + uscite simultanee RGB, Composito, S-VHS + Picture in picture display +Programmi dedicati (GVP scala 24bit, Caligari 24bit, Macro-point 24bit) +Control Panel.

SK 24bit	Sch. Grafica 16.000.000	4.854.000
GVP550	Adattatore per A2000	133.000

ADVANCED STORAGE SYSTEMS NEXUS
SCSI & RAM controllers, e Software di gestione HD per AMIGA2000

Interfaccia SCSI ad alte prestazioni, espandibile fino a 8MB, garanzia 5 anni. Completa di Software di gestione Hard-disk: FlashBack, Powerbench, Smart-cache, Spoollt, Disksurgeon, Instantformat, Memory-doctor

SCSI controller		
40MB	Teac	450.000
53MB	Quantum	849.000
105MB	Quantum	980.000
170MB	Quantum	1.390.000
170MB	Quantum	1.790.000
210MB	Quantum	1.950.000
425MB	Quantum	3.592.000
128MB	Ottico R/W removibile	2.980.000
600MB	Ottico R/W removibile	5.980.000
Cartuccia per 600MB		299.000
Cartuccie per 128MB		99.000
Modulo RAM da 2MB		220.000

COMMODORE COMPUTER

Amiga 500	68000 7Mhz 512Kb	645.000
A500 Plus	68000 7Mhz 1MB	739.000
A500 Ap.	A500 PLUS +Soft. ap.	749.000
	STEINBERG	1.390.000
Amiga 2000	68000 7Mhz 1MB	1.340.000
CD-TV	Riproduttore CD-TV	1.150.000

COMMOCORE MONITOR

1084S	Monitor colore Stereo	450.000
1950	Monitor alta risoluzione colore Multisync	695.000
A2024	Monitor alta definizione 4 grigi per DTP	840.000

COMMODORE MISC PRODUCT

A590	HD 20MB per A500 espandibile a 2MB RAM	639.000
A520	Modulatore TV	49.000
A2088	Scheda Bridgeboard Janus XT	610.000
A2086	Scheda Bridgeboard Janus AT	839.000
A2300	Genlock per Amiga 2000	289.000
A2320	De-interlacer Flicker fixer	390.000
A10	Altoparlanti stereo amplificati per Amiga	69.000

HARDWARE AMIGA

ACD 68040 FUSION FORTY
Scheda acceleratrice per AMIGA 2000

Motorola 68040 a 25Mhz, 25MIPS, espandibile 4MB, 16MB o 32MB RAM a 32bit

68040RAM 4MB	4.890.000

ICD AdScsi2080
SCSI controllers + RAM peer AMIGA 2000

SCSI controller		299.000
40MB	Teac	698.000
50MB	Quantum	829.000
105MB	Quantum	1.239.000
170MB	Quantum	1.639.000
210MB	Quantum	1.799.000
425MB	Quantum	3.441.000
Modullo RAM da 2MB		220.000

ICD AdIde
AT controller per AMIGA

Interfaccie AT-Bus per AMIGA, montaggio all'interno del computer, sia per 500 & 2000. La AdIde40 funziona con tutti gli hard-disk standard AT, mentre la AdIde44 si usa con gli hard-disk da 2,5 pollici.

AdIde 40		319.000
AdIde 44		359.000
Novia20i	HD 20MB int. A500	890.000
Prima52i	HD52MB int. A2000	849.000
Prima105i	HD105MB int. A2000	1.259.000

ICD AdRam2000
Espansioni di memoria per AMIGA 2000

RAM controller	214.000
2MB	390.000
4MB	566.000
6MB	742.000
8MB	918.000

ICD AdRam540
Espansione di memoria da 0 a 6MB per A500

RAM controller	238.000
1MB	326.000
2MB	414.000
4MB	590.000
6MB	1.165.000

ICD Misc Products

AdSpeed acceleratore per tutti i computer AMIGA, 16Mhz e 32Kb di cache-ram a 32bit. Flicker free video per tutti gli AMIGA, alta qualtà senza flickering.

AdSpeed	460.000
Flicker Free	690.000

POWER RAM
Espansioni di memoria per AMIGA 500

512Kb	no clock card	69.000
512Kb	clock card	85.000
1.5MB	clock card	239.000

POWER DRIVE

Drive esterni ed interni per AMIGA 500 & 2000. Il nuovo PC880B ha il nuovo copiatore hardware NewBlitz e l'antivirus integrati su scheda.

PC880	Drive esterno	129.000
PC880B	Drive esterno	149.000
DDriverB	Doppio drive	249.000
PC882	Drive int. per A 2000	115.000
A500D	Drive int. per A 500	115.000

POWER PERIPHERAL

Mouse	optomeccanico 290dpi	49.000
Mouse	ottico 300dpi	99.000
Trackball		75.000
Midi	interfaccia	44.000
Microw	Flicker Fixer per A2000	399.000
Datel	Action replay A500	159.000
Datel	Action replay A2000	169.000
Scanner	con Soft di gestione im.	385.000
AT-ONCE	Emulatore MS-DOS	369.000
AT-ONCE	Ad. AT-ONCE A 2000	164.000
Powerboard	Emulatore MS-DOS con esp. 512 e DOS originale	590.000
NEWBlitz	Copiatore Hardware + antivirus	59.000
Mat	Tappetino per il mouse	13.000
Opt. Mat	Tappetino per m.ottico	25.000

Come Ordinare:
Per Telefono:
Chiamando il 06/5646310 (2 linee R.A.)
Per Posta:
Indirizzando a POWER COMPUTING Srl
Via delle Balear, 90
00121 Ostia Lido - ROMA
Per Fax:
Al numero 06/5646301
Vendita diretta al pubblico:
In Via delle Baleari, 90
00121 Ostia Lido - ROMA

E' possibile pagare con
CARTA DI CREDITO
anche telefonicamente.
SPEDIZIONI IN CONTRASSEGNO
IN TUTTA ITALIA
TUTTI I PREZZI SI INTENDONO
IVA INCLUSA

Tutti i prodotti dove non diversamente specificato sono coperti da garanzia di 12 mesi

Cavetteria per AMIGA & ATARI

VASTO ASSORTIMENTO
SOFTWARE
PER AMIGA & ATARI

DISPONIBILI TUTTI I TITOLI
PER CDTV

HARDWARE AMIGA

POWER HARD-DISK

900E	40MB Slimline	747.000
900E	100MB Slimline	1.229.000
900B	40MB clock	962.000
900B	40MB noclock	923.000
900B	100MB clock	1.389.000
900B	100MB noclock	1.349.000

POWER DRIVE

PC720E	Drive esterno alimentato da Joystick	138.000
PC720	Drive esterno con alimentatore indipendente	188.000
ATARID	Drive interno ATARI senza modifica	118.000

POWER RAM
Espansioni di memoria per ATARI

RAM2	Espansione di memoria per ATARI da 2MB	396.000
RAM4	Espansione di memoria per ATARI da 4MB	586.000

POWER PERIPHERICAL

Mouse	optomeccanico 290dpi.	49.000
Mouse	ottico 300DPI	99.000
Trackball		75.000
Scanner	con Software di gestione immagini	469.000
Blitz	Copiatore hardware	49.000
AT-ONCE	Emulatore MS-DOS 16Mhz	559.000
Ultimate Ripper		89.000
AdSpeed	scheda velocizzatrice	460.000

ICD CONTROLLERS

AdSCSI Micro	controller interno per ST	165.000
AdSCSI Norm	controller per ST	210.000
AdSCSI Plus	controller con clock per ST	215.000

ANNO 5 • NUMERO 34 • MAGGIO 1992
Foto di copertina: © 1990 Pat Lawler - Siggraph '90

POSTA

TRENDS

RUBRICHE

ON DISK

INSERTO

TRANSACTION

RECENSIONI

GAME SHOW

ANCORA *2.0*

Spettabile redazione,
il motivo per cui vi scrivo è il seguente: nel numero 28 (Novembre 1991) leggo a pag.16 "Buone notizie per gli utenti Amiga: la Commodore Italiana ha attivato la campagna 2.0..." e a pag.19 nello speciale SMAU "Amiga 500 Plus e OS 2.0..."
Non vedo il nesso della buona notizia, quando mi si prospetta una spesa e considerando che un mio amico giorni addietro ha acquistato un Amiga 500 Plus e abbiamo constatato che pochissimi programmi interagiscono; quindi nel caso mi decidessi a montare le nuove ROM 2.0 che fine faranno tutti i programmi che ho comprato (non piratati).
Capisco che bisogna sempre guardare in avanti e migliorarsi; però, perché non si fa questo senza creare disagi all'utente?
Spero in tutti i modi di avere un vostro tanto atteso riscontro in merito. Posso, nelle condizioni elencatevi prima, montare le ROM 2.0? Si riuscirà a poter utilizzare tutti i programmi che utilizzavo col vecchio sistema?
Ringrazio anticipatamente, distinti saluti.
Corrado Gargano - Noto (SR)

Caro Corrado,
potrai montare le ROM 2.0, ma non aspettarti, come hai constatato, che tutto il software in tuo possesso funzioni regolarmente. Non pensare comunque che la responsabilità sia da imputare alla Commodore: se i programmatori seguissero maggiormente le direttive della casa madre a quest'ora non staremmo affrontando questo problema! Una cosa però ci sfugge: se sei un utente novizio, come affermi nella tua lettera, perché questa impellente necessità del sistema operativo 2.0? Le case continueranno ancora per anni a supportare i vecchi sistemi e in ogni caso il software disponibile è talmente numeroso e vario da non poterti creare problemi di alcuna sorta. Inoltre, se hai veramente gli originali saprai che molte software house spediscono frequentemente upgrade dei propri programmi e, ultimamente, la compatibilità con il 2.0 e con i processori più veloci è la prima cosa a cui pensano. Per concludere, la spesa da affrontare per le ROM 2.0 è quasi ridicola (si parla di circa lire 150.000 per il Kit), quindi perché questo accanimento? Il futuro non è poi cosi grigio...

MIDI E BASI MUSICALI

Spettabile Gruppo Editoriale Jackson,
sono un vostro lettore della rivista Amiga Magazine. Posseggo un computer Amiga 500 corredato anche di interfaccia MIDI. Nello Spazio MIDI che voi dedicate alla computer music ho molto apprezzato le novità per quanto riguarda i moduli Roland e i vari programmi musicali che avete presentato.
Poiché mi sono affacciato da poco alla computer music, perché prima mi limitavo solo a suonare le tastiere senza alcun ausilio di computer, desidero sapere se per Amiga 500 esistono basi musicali complete come quelle che sono in commercio per l'Atari già da anni. Poiché penso che Amiga non ha niente da invidiare all'Atari (ST N.d.R.) suppongo che esistano. Gradirei, sempre se possibile, ricevere una lista dei brani in caso positivo.
Ringraziandovi anticipatamente della vostra cortesia e scusandomi del disturbo, vi invio cordiali saluti.
Fabrizio De Carlo - S. Giovanni in Fiore (CS)

Caro Fabrizio,
sappi che se il software di sequencer utilizzato con Amiga è in grado di ricevere file in standard MIDI file, allora è possibile caricare su computer tutte le basi musicali su floppy disk da 3 pollici e mezzo ed essendo moltissime ci è praticamente impossibile farne un elenco in queste pagine. Un gran numero di produttori si serve di inserzioni su riviste specializzate in campo musicale, piuttosto che su mensili informatici. Se vuoi un consiglio dai un'occhiata a SM Strumenti Musicali sempre del Gruppo Editoriale Jackson: lì dovresti trovare tutte le basi di cui hai bisogno.

600 SI', *600 NO*

Spettabile redazione,
ho letto il numero di Aprile della vostra rivista e con molto stupore e interesse e, diciamo pure, con una notevole rabbia mi sono tuffato sull'articolo riguardante il nuovo 'gioiello" di casa Commodore. Ma come? E' da poco uscito il 500 Plus e la casa più famosa del mondo ci propone un clone con la scusa del restyling, del controller per Hard Disk e della maggiore compattezza dei componenti! Ora mi chiedo cosa dovrebbero fare tutti coloro che hanno appena acquistato un Plus: mangiarsi le mani per non aver aspettato un nuovo modello (giunto fra l'altro del tutto insapettato) o fregarsene visto che si tratta più o meno del medesimo computer? Non vorrei che si ripetesse la burla del C16 e del Plus 4...
Matteo Bossi - Ferrara

Distinta Redazione,
sono un felicissimo possessore di Amiga 2000, ancora più contento dopo aver letto il numero di maggio di Amiga Magazine, la migliore rivista del settore a mio giudizio (grazie, troppo buono N.d.R.) e l'articolo sul 600. Finalmente la Commodore ha saputo dimostrare di non dormire sugli allori come in passato e vivere di rendita grazie a sporadiche idee geniali (C64 e Amiga 500); il nuovo 600 può conquistare un'altra fetta di mercato contribuendo all'amplia-

mento del mondo Amiga in tutto il mondo e, attenzione, a non fare concorrenza al Plus. Nonostante le similitudini infatti le due macchine sono destinate a due fasce di mercato diverse (come d'altronde spiegava l'articolo in questione) e quindi non dovrebbero darsi fastidio a vicenda. Complimenti Commodore, una mossa davvero azzeccata.
Aldo De Biagi - Napoli

Le due lettere si commentano da sole. Vorremmo solo aggiungere che è inutile pigliarsela tanto: se vi serve una macchina dal costo contenuto, semiportatile ma inespandibile "tuffatevi" sul 600, altrimenti scegliete il Plus.
Certo, così il criterio di scelta è piuttosto semplificato, ma è solo per far capire che le due macchine non si annulleranno a vicenda secondo il nostro parere e le periferiche per Amiga continueranno a fioccare come accade ora.

AMIGA ABARTH

Distinta redazione di Amiga Magazine,
seguo la vostra rivista con notevole interesse perché la ritengo l'unica valida in questo campo, ma avrei un appunto da farvi: la maggior parte delle vostre recensioni è dedicata solo a utenti professionisti o con molto denaro. Molti degli articoli si concludono con frasi del tipo "indispensabile per i posessori di 68030" oppure "il prdotto XYZ funziona, ovviamente, solo sotto Kickstart 2.0". Ovviamente? Eppure credevo che la maggioranza di utenti Amiga possedesse un Amiga 500 espanso a 1 Mb, con drive esterno e basta. Qual è la percentuale di possessori di schede acceleratrici, hard disk e espansioni da 5 Mb? Davvero minima, a mio giudizio. Quindi credo che molti lettori preferirebbero più articoli e recensioni dedicate a una fascia d'utenza più bassa.
Spero che non vogliate intendere la mia lettera come un insulto, ma come critica costruttiva utile a voi come a noi lettori.
Distinti saluti.
Max Vecchi - Roma

Caro Max, tanto per cominciare precisiamo una cosa: le recensioni di Amiga Magazine sono destinate a tutta l'utenza Amiga e proprio per questo motivo nelle conclusioni cerchiamo sempre di trarre un giudizio rivolto sia al possessore di un 500 inepanso che al professionista "armato" di 3000 e Hard Disk da 1.2 Gigabyte.
Ovviamente se l'articolo riguarda un prodotto per il ray tracing non possiamo esentarci dal consigliarlo ai soli possessori di Amiga velocizzati e neppure possiamo esentarci dal recensirlo, poiché le persone interessate a questo argomento sono moltissime, sia che possiedano un 68030, sia che stiano per comprarne uno, sia per curiosità.
Inoltre, sarà vero che l'utenza Amiga è composta (purtroppo) in larga parte da ragazzini con 500 inespanso e joystick collegato come unica periferica, ma non bisogna dimenticare che questa rivista è dedicata a tutti coloro che vogliono sfruttare Amiga in tutte le sue potenzialità. Perciò ecco motivate le recensioni di hardware, di utility, ecco il perché di TransAction e di Game Show che, anche se limitato a due pagine, vuole offrire una panoramica sui migliori giochi usciti per Amiga (perché Amiga vuol dire anche gioco).
Leggendo le vostre stesse lettere ci siamo resi conto che molti di voi possiedono schede acceleratrici, emulatori PC, espansioni di memoria e hard disk.
E ricordate che non siete obbligati a comprare tutte le periferiche che escono quotidianamente per Amiga: Amiga Magazine serve proprio per informarvi e aiutravi a scegliere, sia che abbiate un A500 con 512K di FastRam, sia che possediate un CDTV, sia che lavoriate con un 3000 a 25 MHz. Il mondo è bello perché è vario...

BREVEMENTE

Cominciamo questo mese con la lettera di Antonio Lionello che ha fondato un club per utenti Amiga basato principalmente sullo scambio di trucchi, consigli e altro per l'Action Replay, spesso considerata ingiustamente come la via più breve per piratare giochi.
Per informazioni scrivete a: Lozer Marina - Via Desenzano 2/9 - 30038 Spinea (VE). A tal proposito vorremmo ricordare che su uno dei prossimi numeri ci occuperemo in maniera più approfondita della MK III, terza versione della sopracitata cartridge.
Sì, caro Alessandro Assi di Milano, ad Amiga Magazine leggiamo tutte (proprio tutte) le lettere che ci arrivano e il criterio di selezione delle lettere pubblicabili è semplicissimo: prima vengono scelte le missive con problemi comuni a più lettori e poi quelle interessanti per qualche ragione.
Ovviamente non possiamo rispondere a TUTTE le lettere (non abbiamo ancora più di 40 pagine a numero a disposizione solo per la Posta) quindi non possiamo stampare la tua missiva e nemmeno quelle di altri lettori che chiedono quanti grammi pesa una certa scheda o domande sul genere (e, credeteci, ne arrivano svariate decine simili!).
Sono arrivate le prime risposte circa l'annoso problema dei demo sul dischetto di Amiga Magazine e, nove volte su dieci, sono decisamente contrarie, quindi per adesso niente demo sul disco allegato!

PRECISAZIONI

All'ultimo momento, l'AP&S, distributrice italiana del DCTV, ci ha comunicato che, contrariamente a quanto affermato nell'articolo a pag. 59 di questo numero, il DCTV in commercio non contiene la versione PAL del programma DCTV (presente nel disco 1). Il programma comunque funziona perfettamente sotto Kickstart 2.0 mentre l'utente di Kickstart 1.2 o 1.3 dovrà quindi usare i due programmi separati DCTVPaint e DCTVProc, equivalenti dal punto di vista funzionale al primo. Infine, il prezzo previsto sarà di 1.090.000 lire IVA compresa.

DALLA STAMPA DI TUTTO IL MONDO

IL MONDO ALLA ROVESCIA

Sul numero di aprile della rivista tedesca Amiga Magazin compare la pubblicità del Macintosh Classic: l'avviso cerca di far leva sul basso prezzo d'acquisto del prodotto.
La rivista, come sapete, è dedicata esclusivamente ad Amiga, che in Germania domina letteralmente il mercato consumer.
Un'Apple che fa la corte al mondo Amiga indubbiamente sorprende, ma testimonia, probabilmente, la crescente importanza del mondo dell'home computer nel contesto del volume complessivo del mercato informatico.

UNA SCHEDA DIGITALIZZATRICE A 24 BIT

La tedesca Macro System, nota per i suoi hard disk Evolution e MultiEvolution, nonché per la scheda audio Maestro, pubblicizza una nuova scheda interna per A2000 e A3000, capace di catturare immagini a 24 bit ad alta velocità. La pubblicità accenna anche alla possibilità di PIP e al supporto ARexx.
Si chiama VLAB e il prezzo previsto in Germania è di 598 marchi.

UN CONTROLLER HARD DISK AD ALTA VELOCITA'

La Masoboshi ha annunciato un nuovo controller DMA, AT e SCSI che consente anche di aggiungere fino a 8 Mb di RAM (Zip): la scheda è autoconfigurante, automount e RDB compatibile. Ne da notizia la tedesca Amiga Magazin, che riporta un transfer rate di 1060 K in letturae di 940 K in scrittura, su 68000.11nome è Mastercard e il prezzo in Germania con hard disk Quantum 52 LPS è di 950 marchi.

GRAFICA A 24 BIT

L'AVideo 24, una scheda grafica a 24 bit, da montare internamente all'A2000 e all'A3000, al posto del Denise (ne avevamo dato notizia tempo fa), viene ora commercializzata in Germania assieme alla più economica AVideo 12 (colori a 12 bit, adatta anche all'A500). Il prezzo previsto sarà di 1300 marchi per il modello maggiore e di 800 per quello più piccolo.
Le due schede sono state ribattezzate ColorMaster 12 e 24, per il mercato tedesco. Ricordo che il prodotto è d'origine francese e non è ancora disponibile in Italia.

REFLECTION 2.0

E' uscita la nuova versione di Reflection: il programma di ray tracing pare essersi profondamente rinnovato. Lo testimonia, se non altro, la politica di upgrade: il prezzo previsto è di 349 marchi, l'upgrade dalla versione precedente costa ben 249 marchi.

DESKTOP VIDEO

La rivista americana Amiga World ha pubblicato un numero speciale (che si affianca alla normale pubblicazione della rivista) dedicato al DeskTop Video e all'animazione.
Il numero è piuttosto interessante, perché offre una panoramica a 360 gradi sull'aggrovigliato mercato

OVER THE TOP

In occasione della manifestazione "Amiga '92". tenutasi a Berlino dal 3 al 5 aprile, la Hardital di Milano ha presentato al pubblico la "OVER THE TOP", una nuova scheda acceleratrice per Amiga 2000 basata sul processore Motorola 68040.
Le caratteristiche principali sono:

- CPU MC68040 che incorpora MMU, FPU e due cache separate da 4 Kb per dati e istruzioni;
- Design asincrono per permettere in futuro, quando disponibili. l'utilizzo di processori a 33 e 40 MHz:
- Un apposito processore gestisce fino a 32 Mb di RAM a 32 bit facendo uso di moduli SIMM da 1Mbit o 4Mbit:
- possibilità di aggiungere un modulo con un controller hard disk in standard SCSI II da 32 bit.

Un deviatore permette la selezione del modo 68040 o 68000 per permettere la compabilità con tutto il software esistente; inoltre, un altro deviatore abilita o disabilita il controller hard disk. Ovviamente la scheda è compatibile con i sistemi operativi '1.3 o 2.0. Infine, il prezzo come ultima sorpresa: nella configurazione base senza RAM sarà di Lire 1.490.000.

Per ulteriori informazioni:
Hardital S.r.l. - Via G. Cantoni, 12 - 20144 Milano
Tel. 02/4983457-4983462 - Fax. 021498346

ABACUS

Dal 7 al 11 maggio si terrà., alla fiera di Milano, la terza edizione di ABACUS, la prima mostra-mercato italiana.
A questa edizione di ABACUS parteciperà la Commodore Italiana con tutte le ultime novità, la Epson con la sua gamma di stampanti e tanti altri interessanti produttori, sia hardware che software. Inoltre, i visitatori potranno entrare nella Casa Intelligente, nel mondo della Realtà Virtuale e fare un Viaggio nel Futuro: insomma un appuntamento da non perdere!

video per Amiga. Tuttavia, il punto di vistaè strettamente statunitense e poco spazio è riservato a prodotti, pur validi, di origine europea, sia in campo hardware, che software (non sono citati né Scala, né l'Harlequin).
Gli articoli, quasi una ventina per un totale di un centinaio di pagine (lapubblicità è molto inferiore alla media di Amiga World), affrontano temi disparati, accomunati solo dal generico riferimento al mondovideo e grafico: le basi del DeskTop Video, le tecniche di editing video, il concetto di time code, la recensione di alcuni Time Base Corrector, quella di tre Single Frame Controller per la registrazione di animazioni su videoregistratore a passo uno e di altri tre prodotti per l'editing video, tecniche per l'utilizzo del Toaster, articoli di vario taglio sulla creazione di animazioni, una panoramica sui prodotti che consentono di combinare immagini e suoni, un esempio di grafica 3-D con superfici riflettenti, una panoramica sui dispositivi a 24 bit (dall'HAM-E alla Rambrandt, ma nessun prodotto europeo).
Una panoramica sui programmi di Paint, da 32 a 16 milioni di colori, l'analisi tecnica comparativa di 10 Genlock, una panoramica sulla titolazione video, un glossario 3-D e altre piccole cose. Si tratta, tutto sommato, di uno speciale utile che permette di farsi un quadro ragionato e completo (anche se non sempre approfondito sotto il profilo tecnico) dello stato del DeskTop Video: un mondo in rapida espansione negli USA e in cui Amiga sta giocando un ruolo chiave.
La crescita del settore ha spesso proiettato gli utenti Amiga in un universo poco familiare, perché riservato, fino a poco tempo fa, solo agli addetti ai lavori.
Questo numero contribuisce afare il punto nell'unico settore professionale in cui Amiga svolge un ruolo di primaria importanza.

EMULATORE MEGADRIVE

L'elenco di piattaforme hardware emulate da Amiga sta per allungarsi: secondo un messaggio circolato nella rete telematica amatoriale Fidonet, l'inglese Advanced Emulation sta lavorando a un emulatore software-hardware del Sega Megadrive, che dovrebbe consentire di caricare il software Megadrive da floppy, hard disk o lettore di CD-ROM (anche mediante l'atteso A570 delle Commodore).
La società dichiara una velocità pari al 60-65% di quellaoriginalecon il 68000 e del 95% con un 68030. L'aspetto che risulta più sacrificato è il sonoro, mentre i 64 colori della console vengono ottenuti o in modo half brite (ma certi colori non vengono rispettati), o in modo HAM dinamico (ma solo sul 68030).
L'emulatore puòfunzionare con i lettori di CD-ROM della Philips e della Goldstar. Sono necessari 1.5 Mb di RAM e il Fat Agnus.
E' previsto anche un adattore per leggere le cartucce ROM, eseguirle direttamente o trasferirle su disco, mediante la porta parallela.
La versione professionale fornisce anche delle funzioni da Action Replay. Riesce anche a caricare molti dischi in formato giapponese.
La Sega pare non abbia approvato il progetto. L'AmigaDrive Emulator dovrebbe essere disponibile dal 1 Maggio 1992 al costo di 99 sterline per la versione normale e di 150 per quella professionale. Il software per il CD costa 75 sterline e l'hardware per le cartucce altre 20 sterline.
Forniamo questa notizia così com'è: non abbiamo potuto vericare in alcun modo la sua attendibilità e quindi non possiamo escludere che sia del tutto infondata.

Hinter Bringer

NOVITA' MUSICALI

La Biue Ribbon SoundWorks ha annunciato una serie di novità nella sua linea di prodotti musicali. SyncPro (199 dollari) è un programma di livello professionale, completamente configurabile, che permette di sincronizzare audio MIDI e video, mediante gli standard SMPTE, Midi Time Code e Song Pointer Sync. Il prodotto è compatibile con Bars & Pipes Professional e SuperJAM!, ma può essere utilizzato anche con altri programmi audio. E' in grado, fra le tante altre cose, di leggere e generare tutti i formati SMPTE e di convertirli nel formato MIDI Time Code. Tools Without a Home (19.95 dollari) è invece una collezione di moduli per B&P e B&PP realizzata da Chip Mattson e Larry Burger, due utenti professionisti di B&PP; i tool, che estendono le funzioni dei due pacchetti della Blue Ribbon, vengono forniti con i sorgenti, per consentire ai programmatori di modificarli secondo le proprie esigenze. Sono stati anche annunciati nuovi dischi Extras per SuperJAM!. Si tratta di raccolte di nuovi stili musicali, intitolati: "Pop/Rock Disk", "Classical" e "Cutting Edge Disk". Sı possono richiedere, al prezzo di 24.95 dollari l'uno, esclusivamente alla Blue Ribbon.
E' infine stata annunciata la versione 7.0e di Bars & Pipes Professtonal: sono stati corretti alcuni bug e sono state aggiunte funzioni che permettono d~ usare fino a pieno Pro Studio Kit e Creativity Kit. L'upgrade verrà inviato a tutti gli utenti registrati, che ne faranno richiesta, al prezzo di 9.50 dollari

NEL CASO DI DISCHETTO DIFETTOSO

Può succedere che vi siano alcuni dischetti difettosi sfuggiti al controllo elettronico della macchina duplicatricc: nella sfortunata ipotesi in cui vi imbatteste in uno di questi, vi preghiamo di ritornarci il dischetto difettoso che vi sarà immediatamente sostituito con uno efficiente e rispedito a casa tramite stretto giro di posta.

AMIGA BERLIN '92

Favolose novità dalla fiera tedesca.

Carlo Santagostino
& Carlo Todeschini

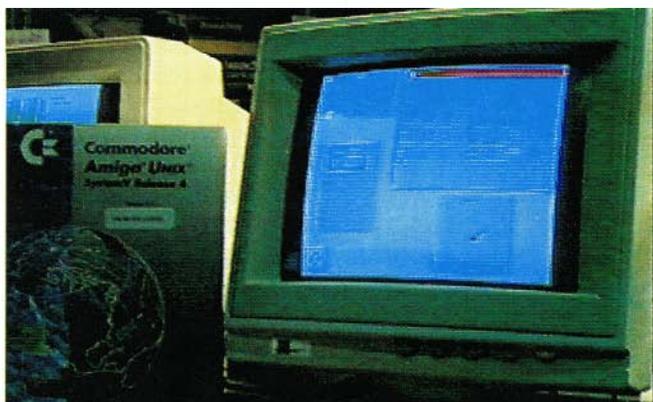
Anche quest'anno si è svolta a Berlino, dal 3 al 5 aprile, quella che fino ad ora è stata considerata la più grande fiera europea (e quindi mondiale) dedicata completamente ad Amiga.
E anche quest'anno vari inviati di Amiga Magazine hanno affrontato il lungo e faticoso viaggio fino alla capitale tedesca, e molto probabilmente per l'ultima volta, dato che le AmiExpo (quella di Colonia e quella di Berlino) non saranno più, a detta di molti, le principali fiere mondiali dedicate ad Amiga. La causa pricipale è che la Commodore non sarà più presente agli AmiShow suddetti, a causa di discordi con l'organizzazione (per quanto abbiamo potuto sentire dai pettegolezzi che giravano) e da quest'anno organizzerà una fiera in proprio che sarà chiamata "World of Commodore" a Francoforte. Noi, chiaramente, saremo anche lì, e vi potremo dire se questo decreterà la fine delle "mitiche" AmiExpo.
Per i motivi sopracitati la quantità degli espositori era notevolmente diminuita rispetto agli anni scorsi, ma le novità non sono certo mancate. Troverete qui di seguito un elenco dei pricinpali espositori con tutte le novità presentate.

W.A.W. ELEKTRONIK

Questa intraprendente società tedesca, presentava, prima in assoluto, un'espansione di memoria specifica per CDTV. E' un'espansione da due Megabyte, porta dunque il CDTV ad avere due Megabyte di Chip e uno di Fast.
Alla domanda del perché fare una espansione per CDTV, mercato abbastanza ristretto anche in Germania, quelli della W.A.W. ci hanno risposto che proprio perché si tratta di un mercato ancora vergine, si sono proposti ai possessori di CDTV come unica prospettiva possibile, almeno per chi vuole utilizzare il CDTV come Amiga dotato di CD-ROM, e non come unità a se stante.
La W.A.W. presentava poi delle altre espansioni più "comuni" per il mercato Amiga, tra cui spiccava un adattatore 2 Mb di Chip RAM per 2000 e 500 molto ben integrato.

TEXT & DATA COMPUTERSERVICE

Anche se questo stand non attirava certo l'attenzione, vuoi per il posto,

L'entrata non era così affollata come gli anni passati, che sia la fine delle AmiExpo?

X-Windows in funzione su Amiga.

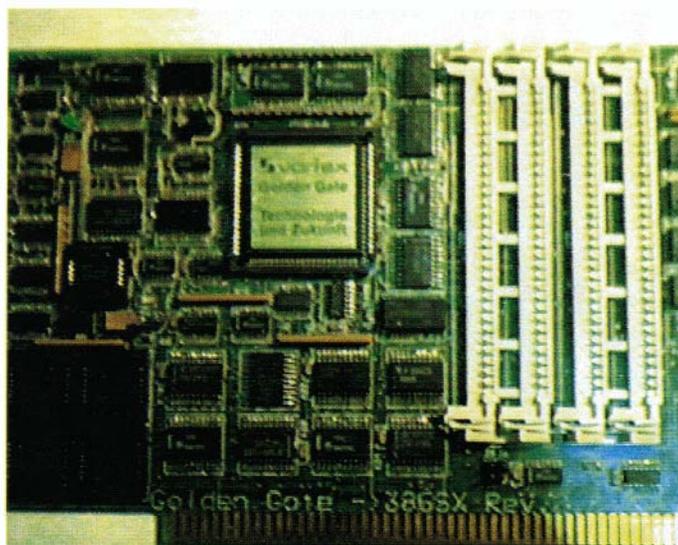
DynaCADD all'opera.

in fondo al padiglione, vuoi per la confusione che regnava; comunque siamo stati attirati dalla presenza di un Amiga 3000 su cui girava indiscutibilmente X-Windows!
Ebbene sì, lo Unix ufficiale Commodore era in vendita già in tutta la Germania.
Siamo stati accolti da un gentilissimo Sandro Coletta (non fatevi ingannare dal nome, anche se di chiara origine italiana, sapeva dire solo pizza e spaghetti in italiano!) che ci ha mostrato nel dettaglio la confezione di Amix: è molto grande (circa 40x40x50 cm), ma il contenuto non delude, contiene due dischetti di installazione e una cartuccia streamer da 150 Megabyte (!) contenente tutto quel che serve per Unix, il tutto accompagnato da ben cinque manuali per l'utente.
La configurazione richiesta è un Amiga 3000 o 2000 con controller 2091, un HD da almeno 200 Mb, 5 Mb di RAM, un tape streamer Commodore da 150 Mb, e, come se non bastasse, si consiglia l'utilizzo di una scheda acceleratrice per quanto riguarda il 2000 e la scheda grafica Commodore A2410, della quale, udite udite, si vocifera finalmente un driver per AmigaDOS (che la Commodore stia preparando la sua versione di WB device independent graphic?).

PHOENIX

Questa società australiana era presente all'Amiga Berlin con un prodotto che i "dimenticati" possessori di Amiga 1000 accoglieranno a braccia aperte, si tratta della "Resurrection" una piastra madre di ricambio per l'A1000.
Adesso, come recita il volantino pubblicitario, c'è la vita dopo la morte per Amiga 1000: Enhanced Chip Set, 2 Mb di memoria, controller SCSI interno, coprocessore matematico opzionale, multi kickstart già su piastra madre, un connettore b2000 standard Zorro, un connettore video-slot perfettamente compatibile con l'omonimo connettore del 2000 e del 3000, nessuno potrebbe chiedere di più.

VORTEX

Forse la più famosa società tedesca di hardware, grazie al suo ATOnce, emulatore AT per Amiga e ST, era chiaramente presente all'esposizione, con una "golosissima" novità, presentava, infatti, in anteprima il nuovo emulatore 386SX per Amiga 2000 e 3000. Il nome è GOLDEN GATE.
Si tratta di una simil-bridgeboard; si collega, infatti, ad un connetore Zorro e un connettore AT presente sia sul 2000 che sul 3000. Ecco le mirabolanti caratteristiche tecniche: CPU 386SX a 25 MHz, zoccolo per 387, RAM da 512 K su scheda, espandibile fino a 16 Mb, sempre su scheda, con moduli SIMM standard; controller floppy per attaccare drive da 1.44 e 2.88, vengono usati comunque i drive Amiga per emulare dei drive standard da 720 K, controller IDE per HD sempre su scheda (!), è possibile comunque utilizzare partizioni di HD Amiga, e, inoltre, possiede tutte le caratteristiche dei precedenti emulatori Vortex per Amiga (emulazione di schede grafiche, emulazione mouse, parallela, seriale ecc.) ed, infine, è chiaramente possibile aggiungere

Golden Gate 386SX, l'alternativa della Vortex alla Janus 386.

qualsiasi scheda pensata per PC o AT nei connettori presenti nel 2000 o 3000. Il prezzo annunciato era di 1298 DM (900.000 lire circa).

INOVATRONICS

Famosa software house australiana, distributrice del favoloso Directory Opus, del cui autore trovate un'intervista in quest'articolo, presentava oltre ai già famosi pacchetti software di cui trovate menzione nell'intervista, anche una periferica piuttosto strana per i possessori di Amiga 500. Si tratta del "HiQ a500 tower", un case predisposto per alloggiare la piastra madre del 500, normale o plus, per trasformarlo in un clone 2000 Tower.
All'Amiga Berlin era presentato con all'interno un Video Toaster e una GVP 030 Combo, e il tutto funzionava perfettamente! Il prezzo è di $699.95, può essere un'utile alternativa, per i possessori di Amiga 500, ad Amiga 2000.

GOLEM

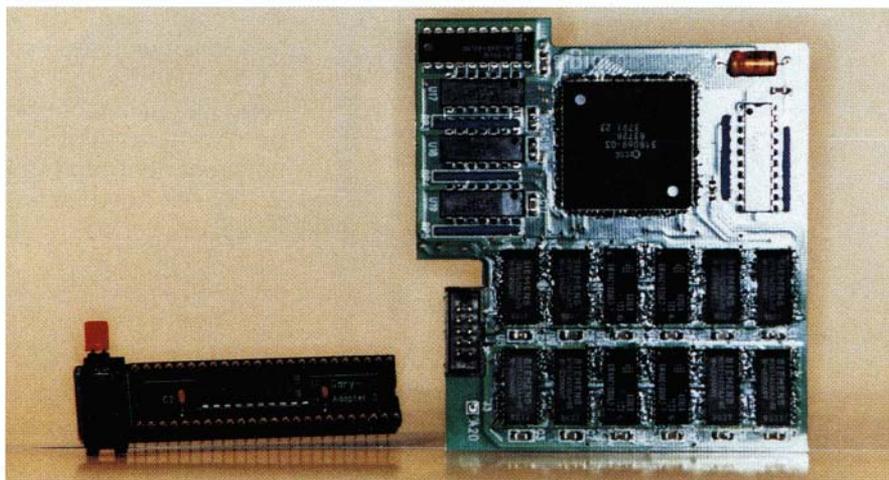
Con uno stand tutto tedesco, la Golem, famosa ditta produtrice di hardware, presentava una quantità spaventosa di suoi prodotti, si andava dalle schede acceleratrici (030 fino a 50 MHz) ai track display per drive. Il tutto però era pensato per il mercato tedesco, dai manuali, al software di installazione, tutto rigorosamente in tedesco. Peccato, perché i prezzi erano veramente bassi, speriamo che in futuro questa ditta si interessi anche al mercato internazionale, cosicché potremo collaudare anche l'affidabilità dei suoi prodotti.

No, non è un 3000 tower, è "HiQ A500 Tower":

ROSSMOLLER

Con uno stand tutto luci e paillettes, era impossibile non notare lo stand della Rossmoller. Presentavano schede acceleratrici con 68000 a 16 MHz, di costo contenutissimo. Un'alternativa alle costose schede basate su 030 o 020 per chi non ha molti soldi da spendere. Un altro prodotto interessante era il Videobackup, interfaccina per collegare il videoregistratore ad Amiga e utilizzarlo come unità di backup, anche questa dal costo molto contenuto (190 DM) veniva venduta con in omaggio una videocassetta contenente i primi 200 Fish Disk. Ultimo prodotto interessante era il Megadrive, che non ha niente a che vedere con l'omonima famosa console della Sega, ma si tratta di un drive da 1.52 Mb esterno per tutti gli Amiga (199 DM).

CRP

Alla CRP, produttrice del DynaCADD, non ci saremmo aspettati di trovare niénte di eclatante, invece ci siamo trovati di fronte ad una vera e propria novità bomba: il DynaCadd/3! Mostratoci da Raul B. Heiduk, che, nonostante il nome non lo faccia presagire, parla benissimo l'italiano, ci ha fatto realmente "strabuzzare gli occhi". Ha tutto, dal Rendering tipo Renderman Pixar (importato dal Silicon Graphics) all'animazione con accelerazioni. Non basterebbe un articolo intero per poter parlare dettagliatamente di tutte le innovazioni che introduce quello che molto probabilmente sarà il programma definitivo di CAD, non solo per Amiga, dato che DynaCADD/3 esiste già per Atari ST e sono previste versioni per MAC, NEXT e PC sotto Windows (anche se per quest'ultimo si sono incontrati dei notevoli problemi nell'implementazione). La novità comunque sicuramente da segnalare è che DynaCADD/3 non sarà un programma di CAD, ma un vero e proprio linguaggio di programmazione per gestione di poligoni nello spazio, ciò significa che il CAD è solo un esempio di quello che si può realizzare utilizzando DynaCADD; pensate alle innumerevoli possibilità che avete, potrete aggiungere le opzioni che desiderate al programma, op-

L'espansione di memoria da 2Mb per CDTV.

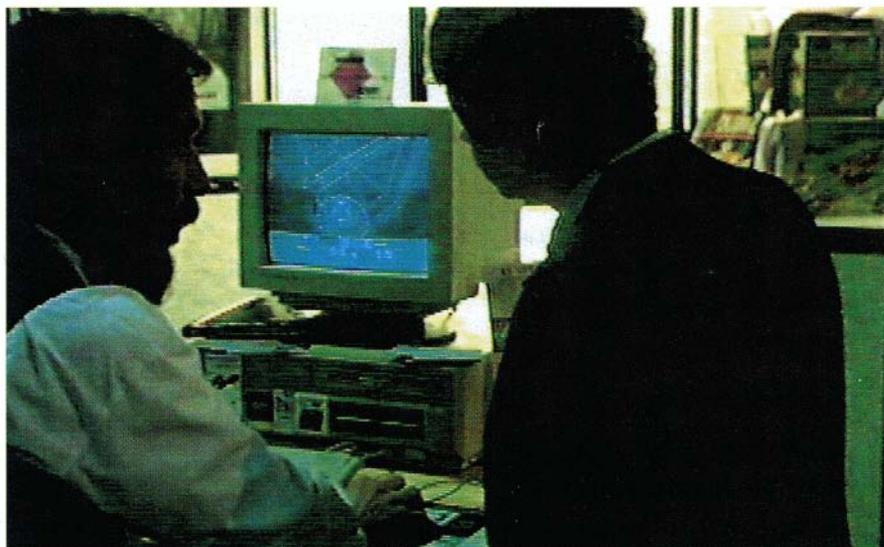
Il nuovo Caligari non ha mancalo di entusiuamare allo Stand della Aeon.

pure potrete addirittura realizzare un programma che non abbia niente a che vedere con il DynaCADD originale, tipo un programma per il disegno strutturato, e questo, indipendentemente da che computer possediate, potrà essere riportato su qualsiasi altro computer con DynaCADD/3!
Vi lascio "in fervida attesa" per un'eventuale news dedicata a questa nuova versione di DynaCADD o addirittura ad una sua recensione.

HARDITAL

L'Italia era presente all'Amiga Berlin, grazie all'Hardital. Questa famosissima società italiana produttrice di hardware sembra godere in Germania una popolarità comparabile, se non superiore, a quella che gode in Italia.
Lo stand era sempre affolatissimo, e molta gente si fermava ad ammirare le schede acceleratrici Big Bang e Super Big Bang prodotte dalla Hardital stessa. Notevole stupore destava anche la presenza in anteprima della scheda 68040, che, stando al programma di Pubblico Dominio Sysinfo, sembra la più veloce presente sul mercato! Aspettiamo con impazienza le schede per una sicura recensione su queste stesse pagine.

COMMODORE

Ma non avevamo detto che la Commodore non era presente? Certo, però i nuovi prodotti Commodore erano presentati da un'infinità di negozi tedeschi presenti all'esposizione.
Innanzitutto il 600 faceva la parte da leone come nuovo modello Amiga, era presente in quasi tutti gli stand, ma non ci dilungheremo molto su questo nuovo prodotto, dato che troverete la recensione completa su questo stesso numero di Amiga Magazine.
Mi dilungherò invece sulla nuova, udite udite, Bridgeboard 386SX Commodore !!!
Già, finalmente la nuova bridgeboard vede la luce, e non delude certo le aspettative: fino ad 8 Mb di RAM su scheda, 386SX a 20 MHz, zoccolo per 387, utilizza i drive Amiga come drive PC, controller drive su scheda per drive da 1.44, ma quello che sorprende di più è il prezzo: 948 DM (circa 700.000 lire!).
Il tutto ingegnerizzato benissimo, pensate, non è una scheda sandwich come la 286, ma singola come la 8086! Ciò vuol dire più spazio all'interno di Amiga 2000 e 3000 e possibilità di inserire più schede AT o XT. La risposta definitiva per chi desidera la compatibilità MS-DOS, aspettatevi al più presto una recensione!

SUPRA

Appena entrati nel secondo padiglione dell'Amiga Berlin '92 era impossibile non notare lo stand circolare dove erano presentati tutti i nuovi prodotti Supra. Oltre ad i soliti controller per hard disk per 500 e 2000, le espansioni RAM e i drive, veniva presentata in anteprima la nuova linea di modem Supra.

La scheda Resurrection per Amiga 1000.

Si tratta di tre nuovi modelli denominati SupraFAXModem che, oltre ad essere ben più veloci dei precedenti, come dice il nome hanno la facoltà di poter inviarelricevere fax.
Il modello di punta, il V.32bis, permette di inviare/ricevere fax e dati ad una velocità pari a 14400 bps mantenendo le caratteristiche di compressione dei dati dei modelli precedenti (MNP 2-5 e V.42bis).
Gli altri modelli erano il V.32 (fax e dati a 9600 bps) e il Plus (fax a 9600 e dati a 2400).
La novità veramente eclatante di questi modem è però il prezzo: pensate solo 869 DM (un po' di più di 600.000 lire) per il modello di punta, 669 DM per il modello a 9600 e solo 399 DM per il più piccolo!

GOLDVISION

In un piccolo ma attrezzatissimo stand era presente la GoldVision, intraprendente software house di Berlino che proponeva un programma di disegno vettoriale veramente potente. Il nome è ExpertDraw ed è stato pensato esplicitamente per la creazione di clip per programmi di DTP. Può lavorare tramite un sistema di retinature veloce e potente in 256 colori, permette la rotazione di caratteri attorno a curve ed effetti di metamorfosi degli oggetti vettoriali. L'output è possibile su stampanti ad aghi presenti nelle preferences, plotter HPGL, postscript, stampanti HP-laser compatibili e, inoltre, può salvare le clip in ProDraw, EPS e, ovviamente, in formato ExpertDraw. Il prezzo è di circa 450 marchi. Di notevole interesse anche una vastissima libreria di font e ClipArt per PageStream e un programma veramente utile che permette di stampare qualsiasi file IFF su un plotter HPGL, anche di quest'ultimo abbiamo avuto una dimostrazione in cui ci siamo potuti meravigliare della velocità con cui il programma ha completato la conversione del file IFF,e della fedeltà del disegno riprodotto dal plotter dal sorgente disegnato da noi in Deluxe Paint.

I FAXmodem della Supra.

AEON VERLAG & STUDIO

Passando distrattamente tra gli stand si veniva immancabilmente calamitati da un monitor dove si potevano ammirare animazioni tridimensionali da lasciare a bocca aperta, e si rimaneva ancor di più sorpresi quando si scopriva, guardandosi attorno, che le incredibili animazioni erano state realizzate con la nuova versione del famoso Caligari, uno degli storici programmi di Rendering per Amiga.
Rendering fino ad 8000x8000 pixel, esce direttamente su HAM-E, DCTV fino alle più costose schede come Harlequin e Impact Vision, e queste sono solo le più marginali delle novità presenti in Caligari 2.1, che vedremo in dettaglio prossimamente! ▲

INTERVISTA A JONATHAN POTTER

Durante la recente visita all'Amiga Berlin '92 siamo stati informati della presenza di un personaggio molto noto per le sue produzioni software in ambito Amiga: Jonathan Potter. Ne abbiamo approfittato per una breve intervista riguardo le sue produzioni e i progetti per il futuro. Per chi non avesse ancora focalizzato il personaggio, ricordiamo che Potter è australiano ed è il creatore di Directory Opus, il piu potente e flessibile "directory utility" attualmente presente sul mercato, nonché di famose utility PD come ZeroVirusIII e WorldTime.
Lo abbiamo potuto incontrare allo stand della INOVAtronics, una ditta americana di cui parleremo in seguito, che ha acquisito i diritti per la distribuzione del programma Directory Opus che, se ben ricordate, non è mai stato di pubblico dominio.
La prima impressione che abbiamo avuto di Potter è stata di sorpresa in quanto si distacca di netto dallo stereotipo del programmatore professionista che ci fa immaginare un ometto magro, alto e con gli occhiali spessi: infatti ci siamo trovati davanti un omone di notevole stazza, senza occhialoni e con una simpaticissima faccia da buono (vedi foto).
Ci siamo subito presentati e, dopo i complimenti di rito per il suo programma, siamo andati subito al sodo. Ci ha spiegato che la collaborazione con la INOVAtronics gli permette di avere una distribuzione e una pubblicità più ampia del suo prodotto nonché un servizio migliore per gli upgrade. Infatti, i vantaggi per gli utenti registrati sono notevoli. Per quanto riguarda il futuro del programma, Potter ci ha detto che per il momento continua sulla strada della compatibilità anche con il DOS 1.3 ma annuncia che a partire da Settembre/Ottobre di quest'anno, Directory Opus sarà completamente 2.0 compatibile: è solo questione di aspettare che il nuovo Kickstart abbia una diffusione più ampia. In questo modo il programma avrà i gadget, le finestre, i requester standard e, cosa più importante, sarà finalmente dotato di gadget di profondità (ce lo siamo fatto assicurare per ben due volte!). Inoltre, ci ha confermato che le nuove versioni saranno ancora più programmabili di quella attuale (non riesco proprio ad immaginare come sia possibile...) e che terranno conto delle nuove funzionalità del Workbench 2.1 come, ad esempio, la "localizzazione" dell'applicazione, ovvero la possibilità di adattarsi alla lingua della nazione dove viene utilizzato il programma.
A questo punto, Potter ha chiamato in causa una persona della sua stessa stazza con un buffo cappello in testa e una somiglianza pressoché perfetta con il comico italiano Carlo Verdone e che stava tranquillamente mangiando un panino all'interno dello stand: si trattava di Eddie Churchill, il programmatore di CanDo, altro prodotto della INOVAtronic. Eddie Churchill ha confermato che anche lui sta lavorando ad una nuova versione del suo programma compatibile DOS 2.0 e cosi, con enorme gioia, possiamo annunciare che spariranno gli orrendi file requester utilizzati da CanDo e il tutto diventerà conforme al nuovo look del sistema operativo e, naturalmente, ancora più potente. Tra un morso e l'altro si è lasciato sfuggire che alla INOVAtronic stanno lavorando per conto della Commodore. Si tratterebbe di una applicazione (forse AppBuilder?) che consentirà ai programmatori di costruire l'interfaccia utente del proprio programma in maniera intuitiva e veloce sullo stile di PowerWindows e di certo si sa solo che sarà qualcosa di eccezionale! Infine, ci ha anche illustrato la nuova versione di Metascope, un potentissimo source level debugger totalmente 2.0 style.
A questo punto, Mister. Potter ci ha donato una meravigliosa spilla a testa che recita: "Jay Miner (padre di Amiga N.d.R.) usa Directory Opus... e così faccio anch'io!".

Potter tenta, senza successo, di decifrare l'italiano.

ESPLORIAMO IL MONDO MIDI (prima parte)

Tutti i segreti del MIDI.

Andrea Laus

Quest'anno il MIDI festeggia il suo ottavo anno di vita e, in tale occasione, le nuove possibilità offerte ai musicisti dall'introduzione del sistema GS Roland lo consacrano protagonista assoluto della musica elettronica. Tuttavia, il MIDI rappresenta ancora, per molte persone, soprattutto per i musicisti che muovono i primi passi nella musica elettronica, un mistero sotto molti punti di vista. Chi non si è trovato, almeno una volta, impacciato di fronte alla necessità di operare collegamenti MIDI multipli fra i vari apparati del proprio studio?

Finché si tratta di una tastiera e di una drum machine le cose sono semplici, ma se si aggiunge al sistema anche un solo expander allora non è più tanto facile collegare tutti gli apparecchi tra loro, via MIDI, nel modo corretto. Un altro problema, altrettanto comune, è quello che si presenta quando si tratta di editare una registrazione. Infatti esistono sequencer (anche abbastanza potenti come il KCS 3.0) che presentano come unico modo per editare quanto si è suonato, la finestra di Event Editor, in cui la musica e i vari comandi sono espressi sotto forma di messaggi MIDI da modificare.

A questo punto, chi non sa nulla di MIDI, di solito, cede le armi, perché non è in grado di dialogare con il computer a questo livello. Si può poi citare il caso del musicista che intende crearsi una workstation e che, privo di conoscenze oggettive relative allo standard MIDI, non dispone di criteri sufficienti per decidere quali strumenti e quali programmi acquistare. E gli esempi potrebbero continuare: accade spesso, infatti, che il musicista si avvicini al mondo MIDI senza conoscerlo e utilizza dei programmi che eseguono operazioni a lui sconosciute, oppure fa uso di strumenti di cui ignora l'architettura, almeno sotto il profilo MIDI. In effetti, non esiste molta letteratura sull'argomento e quella esistente, perlopiù, tratta il MIDI come un argomento da "iniziati". Così, il povero amatore, che magari vuole solo sapere come collegare due expander al sequencer o come editare il proprio brano, si trova, indifeso, ad affrontare manuali spesso incompleti o lacunosi. Naturalmente, non occorre conoscere a memoria tutto il protocollo MIDI per poter lavorare bene con sequencer e altri programmi simili. Tuttavia, credo sia molto utile conoscerne almeno la struttura e le modalità operative, per potersi muovere con maggiore disinvoltura.

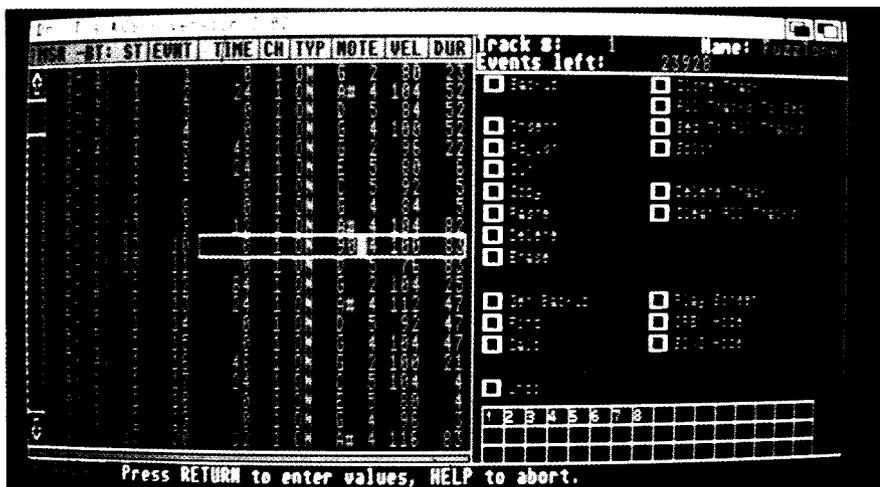



Figura 1: La schermata di Event Editor del KCS 3.0 della Dr.T's.

Anche perché la conoscenza, per quanto limitata, di tale standard mette in condizioni di sfruttare a fondo i propri programmi e la propria attrezzatura musicale. Infine, è assai più gratificante lavorare sapendo cosa succede, invece di lasciare tutto in mano ad un programma.
In questo articolo cercheremo di capire un po' quali siano e come lavorano i messaggi MIDI, prestando particolare attenzione a quelli più comuni, la cui conoscenza è indispensabile per editare un brano. La prossima volta affronteremo, invece, un altro tema spinoso: parleremo dei collegamenti fra varie apparecchiature MIDI, così da fugare tutti i dubbi in proposito. Il tutto, naturalmente, applicato ad Amiga. Infatti, dopo un inizio incerto nelle applicazioni MIDI, oggi Amiga rappresenta un punto di riferimento anche per i musicisti professionisti. Ciò è confermato da decine di programmi applicativi MIDI che coprono tutta la gamma di utilizzazione: sequencer, editor, notazione, composizione, accompagnamento, Editor/Librarian, eccetera. Inoltre, non bisogna dimenticarlo, Amiga rappresenta un faro nella nebbia anche per chi si diletta di applicazioni video, quali le animazioni e, sono felice di dirlo, sono sempre più numerosi i programmi che sposano questi due mondi, quello musicale e quello grafico, tanto che è già possibile crearsi in casa dei piccoli cartoni animati, con la colonna sonora generata direttamente via MIDI.

MIDI: I MESSAGGI FONDAMENTALI

Analizziamo, dunque, la filosofia del nostro standard. L'obiettivo del MIDI è la codifica dell'esecuzione musicale, ovvero catalogare e caratterizzare in modo digitale, secondo un opportuno protocollo, tutte le azioni che il musicista compie sullo strumento (che, per praticità, per il momento, assumiamo sia una tastiera). Cominciamo quindi a vedere come è trattata l'operazione più naturale: la pressione di un tasto per generare una nota. L'invio di una nota è caratterizzato da tre parametri: il numero della nota suonata sullo strumento, il valore della dinamica che viene impartita alla nota e il numero del canale MIDI a cui la nota è indirizzata. Questi sono i tre punti fondamentali su cui si basa la trasmissione e la ricezione di una nota musicale nel sistema MIDI. Ogni nota musicale viene codificata con un numero che va da 0 a 127; naturalmente, nessun strumento musicale possiede una tale estensione, anche perché certe note non sarebbero udibili dal nostro orecchio; quindi, di solito, l'estensione utile va da 21 a 108. Anche la dinamica ha una codifica da 1 a 127, ma questa volta il concetto è quello di volume: se il valore è basso la nota sarà suonata piano e viceversa. Una curiosità: si chiama velocity perché, in realtà, questo parametro misura la velocità di pressione del tasto, e, quindi, il tempo impiegato dalla nota per passare dalla posizione iniziale di quiete a quella finale. I canali MIDI disponibili sono 16, ogni canale è del tutto distinto dall'altro e si potrebbero mandare contemporaneamente 16 informazioni diverse ai 16 canali senza che una interferisca con l'altra. Questo spiega l'importanza del concetto di canale MIDI: ogni canale è un sistema a sé; per fare un esempio, in realtà molto riduttivo, si può paragonare il canale MIDI a una delle tracce di un registratore multipista: avere 16 canali MIDI significa poter avere 16 strumenti diversi che suonano in contemporanea. Ciò, in pratica, è subordinato alle capacità delle singole tastiere che, soprattutto le più economiche, non sempre permettono di trasmettere 16 parti contemporaneamente, cosa invece perfettamente realizzabile mediante un sequencer. Così, ogni volta che suoniamo una nota sulla nostra tastiera e la inviamo via MIDI al nostro sequencer, passiamo queste tre informazioni fondamentali: il numero della nota, la sua velocity (o dinamica) e il canale MIDI; questo primo messaggio viene indicato con il nome di Nota ON. Quando rilasciamo il tasto, mandiamo via MIDI un altro messaggio, che serve appunto a codificare il fatto che quella nota ha smesso di suonare. Tale messaggio, detto Nota OFF, è codificato in modo simile al precedente: restano invariati i parametri relativi al numero della nota e al numero del canale, ma il valore della velocity diventa zero. Naturalmente, sia nel caso di Nota ON che in quello di Nota OFF, il sequencer aggiunge altre informazioni relative, ad esempio, al tempo di esecuzione e al clock, per posizionare esattamente la nota all'interno dell'esecuzione, ma questo, per il momento, non ci interessa. Negli strumenti musicali elettronici ci sono però altri controlli

La schermata di Event Editor del Music-X della Microillusions.

per modificare l'esecuzione secondo le proprie esigenze. Ad esempio, si può intervenire con pedali che svolgano la funzione di sustain o quella di volume, oppure è possibile, tramite un tastierino numerico presente sulla tastiera, cambiare il timbro (preset o, in MIDI, Program Change) con cui si suona. Anche tutte queste informazioni sono state, naturalmente, codificate nel protocollo MIDI. In tal modo, ogni volta che fate una registrazione con un sequencer e modificate qualcosa (dalla pressione della nota al cambio di volume), questo viene registrato dal sequencer. Prima di continuare, analizziamo il comando di Program Change che, assieme a quello di nota, è il più usato in un sequencer. E' composto da due parametri: il numero del canale MIDI (da 1 a 16) su cui si esegue l'operazione e il valore del Program Change da attribuirgli (da 1 a 128).
Anche per questo messaggio, naturalmente, il sequencer aggiungerà altri parametri posizionali, per stabilire quando, effettivamente, esso debba essere inviato.

L'EDITING

In figura 1 si vede la schermata di editing del noto programma per Amiga KCS 3.0 della Dr.T's. Faccio riferimento, per comodità, a questo programma; tuttavia, con minime differenze, dovute solamente all'architettura dei vari sequencer, le operazioni da eseguire in questo tipo di editing sono standard per tutti i sequencer che lo prevedono. La prima cosa che appare è il lungo listato, che si presenta suddiviso ordinatamente in più colonne. Cominciando da sinistra, troviamo le colonne di riferimento per il sequencer: qui viene indicato a che punto della traccia o della sequenza ci troviamo e tali informazioni sono espresse sia sotto forma di Clock MIDI che di step di battuta. Ovviamente, tali informazioni non sono editabili, potete solamente posizionarvi su di esse. Accanto ad esse vengono visualizzate le informazioni più interessanti: innanzitutto il momento, rispetto alla battuta, in cui il messaggio viene inviato. Si tratta di un'informazione molto importante, soprattutto per le note: infatti, suonando in tempo reale, è facilissimo andare fuori tempo e in questa colonna possiamo modificare a piacere il tempo di entrata di una nota, senza interferire su altri parametri. La colonna successiva è quella del canale MIDI, tale informazione, in un sequencer, è fondamentale, dato che quasi tutti i messaggi che utilizziamo qui ne fanno uso. Segue la colonna in cui è specificato il tipo di messaggio. Può essere un messaggio di nota (qui indicato con ON), un messaggio di Program Change (indicato con PC) o un altro messaggio MIDI. Inoltre, in questa colonna trovano posto anche messaggi non MIDI, ma propri del sequencer (ad esempio, il comando per far suonare un'altra traccia); è bene saper distinguere, in fase di editing, tra messaggi MIDI e non, dato che ormai tutti i sequencer li usano entrambi. Dopo il tipo di messaggio, segue il valore ad esso assegnato. Nel KCS questacolonna viene genericamente chiamata Note, dato che, se il messaggio è di nota, qui si specifica quale essa sia; bisogna fare attenzione, però, poiché, se il messaggio è un PC, allora il valore successivo è un Program Change e non una nota, la stessa cosa vale per gli altri messaggi. Il valore della nota non viene espresso qui secondo lo standard MIDI (come un numero, cioè, da 0 a 127), ma, per facilitare le cose, la nota viene indicata con la nomenclatura inglese (A, B, C,..), seguita dal numero dell'ottava; sarà poi il programma ad operare la conversione nel formato MIDI. Tuttavia, dato che non tutti i programmi presentano queste facilitazioni, ho allegato, in figura 2, la tabella delle note con la codifica MIDI. Tornando all'editor, dopo il valore della nota, segue lo spazio per un'eventuale velocity (se il messaggio è di nota). L'ultima colonna ha una funzione particolare, si riferisce solo ai messaggi di nota e si chiama Dur. Il KCS, sempre per facilitare il lavoro del musicista, non visualizza i messaggi di Nota OFF, ma per ogni nota ne indica la durata rispetto all'unità di misura espressa in beat di battuta. E' molto più facile modificare la durata della nota in questo modo, piuttosto che dover inserire nella battuta un messaggio di Nota OFF. In ogni caso, se ci fosse qualche "nota ribelle", è sempre possibile assegnarle velocity zero per spegnerla. Generalmente, il numero dei messaggi MIDI che si editano in un sequencer è molto limitato: desidero però elencarli tutti, in modo che costituiscano, per il musicista, un punto di riferimento. Consiglio poi, nel caso risultasse necessario, di consultare il

NOTA / OTTAVA	DO	DO#	RE	RE#	MI	FA	FA#	SOL	SOL#	LA	LA#	SI
-	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127	-	-	-	-

Figura 2: Elenco delle note musicale con la codifica dello standard MIDI.

manuale del proprio sequencer, per vedere quali sono implementati, in che modo sono gestiti e come editarli.

GLI ALTRI MESSAGGI MIDI

I messaggi codificati dal protocollo MIDI si dividono in due gruppi ben distinti che assolvono funzioni totalmente diverse. Il primo gruppo comprende tutti i cosiddetti messaggi di Canale. Tale nome deriva dal fatto che sono selettivi, non vengono indirizzati cioè a tutti i canali MIDI contemporaneamente, ma a un solo canale. Servono, ad esempio, per mandare un Program Change ad un solo canale MIDI.
Sono previsti due tipi di messaggi di canale: quelli di voce e quelli di modo. I primi controllano le voci (dette anche generatori di nota) di ogni strumento collegato via MIDI a ciascun canale. Essi sono:

- Nota ON
- Nota OFF
- After Touch
- Control Change
- Channel pressure After Touch
- Pitch Wheel Change

Il messaggio di Modo è invece il Channel Mode. Come ho già anticipato, durante l'editing dei dati in un sequencer, ben difficilmente vi troverete a contatto con questi messaggi. Non è quindi il caso di approfondire il discorso. Nel secondo gruppo prendono posto i cosiddetti messaggi di sistema, che non sono subordinati a nessun canale, ma si rivolgono contemporaneamente a tutti i canali e vengono trasmessi e ricevuti immediatamente. Anche questi si dividono in sottocategorie.

- I Messaggi Comuni sono destinati a tutti gli strumenti collegati al sistema, indipendentemente dal canale MIDI (Puntatore brano, Selezione brano, Richiesta accordatura, Fine Sistema Esclusivo)

- I Messaggi in Tempo Reale sono anch'essi ricevuti e trasmessi da tutti gli strumenti collegati al sistema e includono: Impulso di sync, Start, Stop, Continue, Reset e individuazione attività.

- I Messaggi Esclusivi sono stati istituiti per consentire ad ogni costruttore di implementare prestazioni aggiuntive e personalizzate ai propri strumenti: l'elenco di tali messaggi (e dell'implementazione aggiuntiva) si trova nelle pagine del manuale dei singoli strumenti. Esiste, naturalmente, una priorità nell'esecuzione dei messaggi ricevuti, eccola:

1- Messaggio di reset
2- Messaggi di sistema esclusivo
3- Messaggi di sistema in tempo reale
4- Messaggi di canali e di sistema comuni

A questo punto, abbiamo visto tutti i tipi di messaggi MIDI: raccomando ai neo utenti che vogliono imparare ad editare un brano in linguaggio MIDI, di concentrarsi sulla prima parte di questo articolo, quella in cui si trattano i messaggi più comuni (Nota e Program Change) e in cui si descrive il funzionamento di un event editor. Tale tipo di editing, sebbene non sia molto intuitivo, è certamente uno fra i più versatili e con esso è possibile inviare qualunque messaggio MIDI: per questo ritengo sia importante sapere come funziona. Non si spaventino quelli che non riescono proprio ad abituarsi a vedere la propria musica espressa in linguaggio MIDI: la maggior parte dei programmi sequencer implementa il cosiddetto editing a barre, molto più intuitivo da usare e che non richiede una conoscenza così diretta del MIDI come nel precedente. ▲

REAL 3D PROFESSIONAL V1.4

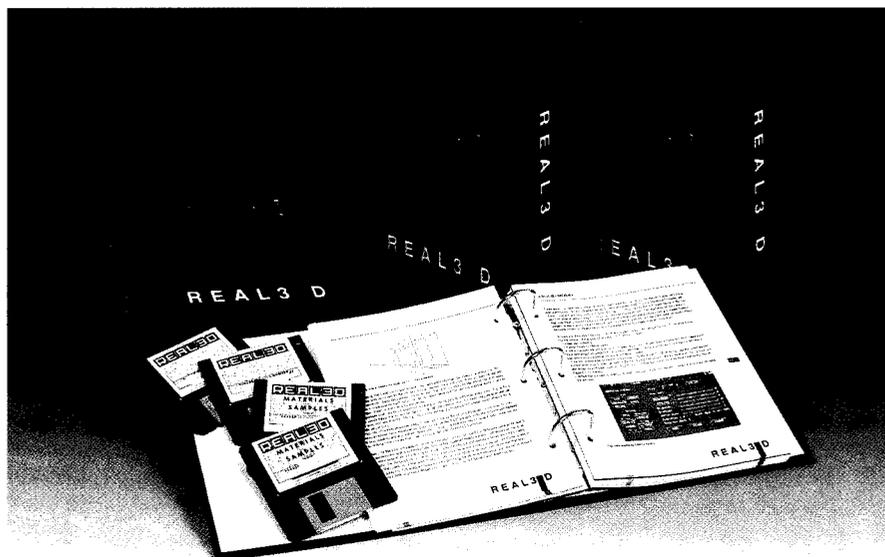
Marco Pugliese

Un fantastico programma per grafica 3D.

Alla fine di quest'ultima stagione invernale, che ha visto l'uscita delle nuove versioni dei più diffusi software di grafica e rendering 3D, quali Imagine 2.0, Caligari 2.0 e LightWave 2.0, ecco che, per ultimo, fa la sua comparsa sul mercato internazionale il nuovo REAL3D giunto ora alla release 1.4.
In un momento in cui non c'è che l'imbarazzo della scelta, questo pacchetto, in virtù delle soluzioni innovative in esso contenute, si pone prepotentemente in evidenza, suscitando la curiosità di parecchi "addetti ai lavori".
Ideato e realizzato da due fratelli finlandesi, Juha e Vesa Meskanen (che insieme formano la Realsoft Ky), REAL 3D fece parlare di sé in passato per il fatto che la versione 1.0 fu il primo software per Amiga a supportare il concetto di operazioni logiche tra oggetti, una caratteristica inglobata poi in quasi tutte le release successive dei programmi concorrenti
Ciononostante la vera novità che distingue questo prodotto da quelli di pari mercato, è sicuramente il fatto che REAL 3D è il primo software di modellazione solida per Amiga che lavori per superfici definite matematicamente (vedi tabella per una descrizione più accurata).
Questa tecnica è un modo molto sofisticato per rappresentare gli oggetti nello spazio tridimensionale; richiede una notevole potenza di calcolo e per questo motivo, fino ad ora, era stata usata solo su computer molto più veloci e costosi di Amiga.
Con l'uscita di REAL 3D, anche per noi è diventato possibile sfruttare i vantaggi di quello che gli autori chiamano "True Solid Modelling", grazie ad un lavoro di ricerca, sviluppo e ottimizzazione che ha impegnato i "nostri" Juha e Vesa per oltre tre anni.
Tale lavoro ha fatto sì che REAL 3D divenisse molto veloce, potente e flessibile, tanto che le cose da dire sono così numerose che comincerei subito a tracciarne un profilo.
Prima di iniziare, devo solo precisare che il programma è giunto in redazione in due versioni differenti: una ridotta e una "Professional".
La versione ridotta (chiamata Beginner), creata solamente per poter vendere il prodotto in due differenti fasce di prezzo, manca di due strumenti: le macro e gli operatori Booleani.

LA CONFEZIONE

La confezione rispecchia pienamente le caratteristiche del suo contenuto: è sobria ed elegante.
Al suo interno trovano posto quattro dischetti e il manuale; quest'ultimo è composto da circa 150 pagine di due differenti colori, rilegate in un raccoglitore ad anelli con copertina rigida; per la versione ridotta i dischetti sono soltanto due e il manuale contiene circa 30 pagine in meno.
Nonostante sia veramente raro che la documentazione inclusa in pacchetto di grafica tridimensionale sia assolutamente esauriente, quella fornita con REAL 3D sembra essere piuttosto chiara e ben fatta; ho ammirato soprattutto la capacità di sintesi che hanno avuto gli autori, riuscendo a descrivere le nume-

L'unica immagine contenuta nella Rel. 1.4. E' un esempio di rendering in Hi-Res a 16 colori.

rose opzioni fornite dal loro programma in un numero abbastanza contenuto di pagine, facendo quindi in modo che chiunque possa mettersi all'opera in tempi molto brevi.

L'INSTALLAZIONE

REAL 3D può essere utilizzato su qualunque Amiga che abbia almeno un Megabyte di RAM e non necessita di drive addizionali; nonostante ciò, come per la maggior parte dei programmi di grafica, sono caldamente raccomandati almeno 3 Mb di memoria e da 3 a 10 Mb di spazio libero sul disco fisso.
Entrambi i pacchetti contengono due versioni del programma: la versione "standard" è per tutti coloro che non posseggono un coprocessore per i calcoli in virgola mobile, mentre la versione "Turbo" è ottimizzata per funzionare sulle macchine più veloci dotate di processori dal 68020 in su, nonché di un coprocessore della classe 6888x.
La procedura di installazione è totalmente automatizzata nel caso venga eseguita su hard disk, mentre va eseguita manualmente (le istruzioni sono comunque chiare) da coloro che purtroppo devono lanciare il programma da dischetto

L'EDITOR TRIDIMENSIONALE

REAL3D è composto da tre moduli principali: l'editor "tri-view", il modo "wireframe", nel quale l'utente può muovere o animare gli oggetti in tempo reale, e il modo "render", dove le immagini vengono effettivamente prodotte.
Una volta lanciato, il programma entra direttamente nell'editor tridimensionale; qui è possibile creare e modificare gli oggetti che compaiono, proiettati ortogonalmente, in tre delle quattro finestre componenti un'interfaccia ormai comune alla maggior parte dei software di grafica 3D (vedi figura 1).
All'interno della quarta finestra però, invece della classica vista in prospettiva, troviamo, oltre a una quarantina di icone che permettono di accedere direttamente ai comandi più utilizzati, una lista a più strati degli oggetti sui quali stiamo lavorando. Essa non è altro che una rappresentazione comoda ed efficiente della gestione object oriented dei solidi che compongono una scena; con REAL 3D infatti, si possono creare oggetti gerarchici, ossia composti ricorsivamente da altri oggetti elementari o complessi. Detto così può sembrar difficile, ma questo tipo di rappresentazione sarà chiaro qualora si pensi a come è organizzato l'albero delle directory su un disco fisso; REAL 3D imita questa struttura utilizzando le sue "directory" per riunire gli oggetti in gruppi e sottogruppi logici i cui nomi, scelti dall'utente, sono appunto visibili nella quarta finestra dell'editor "tri-view".
Un approccio del genere permette di creare e modificare i propri modelli in modo estremamente semplice e veloce poiché, all'occorrenza, è possibile eseguire ogni tipo di operazione sull'intero oggetto oppure su entità logiche separate.
Un'altra importante caratteristica di questo programma è, senza dubbio, il più evidente vantaggio fornito dalla giàcitata tecnica di "True Solid Modelling" utilizzata da REAL3D.
Oltre alle normali superfici piane, è infatti possibile rappresentare un gran numero di superfici curve, come sfere, coni e cilindri, che, a differenza di quanto accade per gli altri software di grafica tridimensionale, non vengono approssimate con poliedri terribilmente complessi o da interpolazioni eseguite durante la fase di rendering, ma sono definite per mezzo di polinomi o, più in generale, da funzioni matematiche.
Ciò significa che è possibile ingrandire una sfera a piacimento, senza che nell'immagine finale compaiano spigoli o sfaccettature non volute.
Una tale costruzione permette, inoltre, di ridurre drasticamente i tempi di calcolo, poiché, come ho detto poc'anzi, vengono evitate tutte le operazioni necessarie per l'interpolazione delle normali (un artificio utilizzato per poter ottenere delle superfici curve anche me-

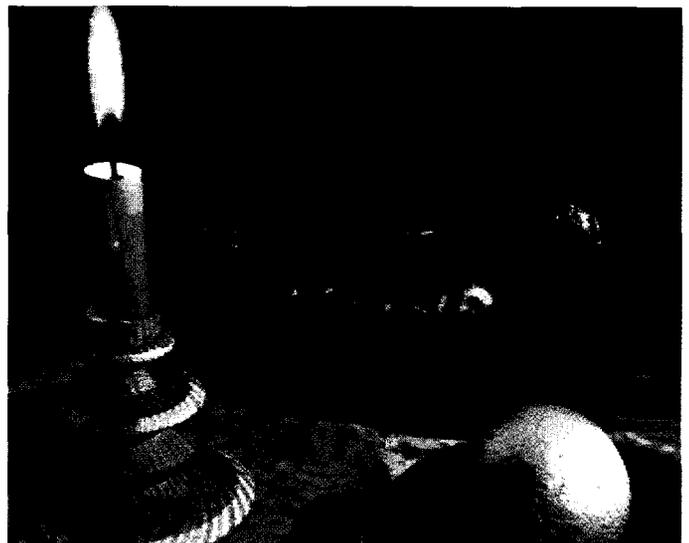
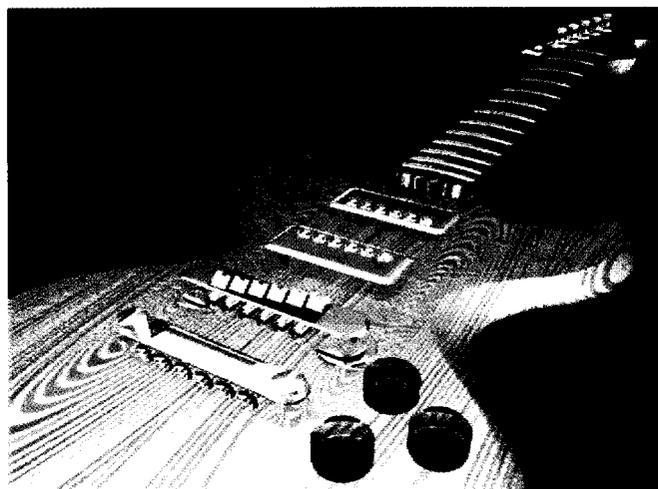
Figura 2: stesso esempio della precedente a colori: si noti il realismo della fiamma.

Esempio di un ottimo utilizzo delle brush.

Figura 1: l'editor di Real 3D.

diante l'uso di soli poligoni). Molti di voi ricorderanno però che le vecchie versioni di REAL 3D, proprio per via di questa loro particolare tecnica di descrizione delle forme solide, risultavano poco flessibili qualora si volesse modellare un oggetto privo di assi di simmetria, modificando, seppure di poco, una primitiva o una sagoma di rotazione; bene, questo limite non esiste più, poiché una delle maggiori novità che questa release 1.4 presenta è proprio la possibilità di lavorare direttamente sui punti di una superficie, quasi come avviene in un programma che lavora per poligoni (come Sculpt o Imagine).

Ora è quindi possibile produrre (anche se in modo non troppo semplice, per la verità) forme estremamente complesse e irregolari, spostando i singoli punti o utilizzando alcuni tool che, combinati fra loro, permettono di deformare gli oggetti in ben 24 modi differenti.

Considerazione a parte, inoltre, merita il potente operatore "Remap", che permette di modificare il numero di punti che definiscono una superficie.

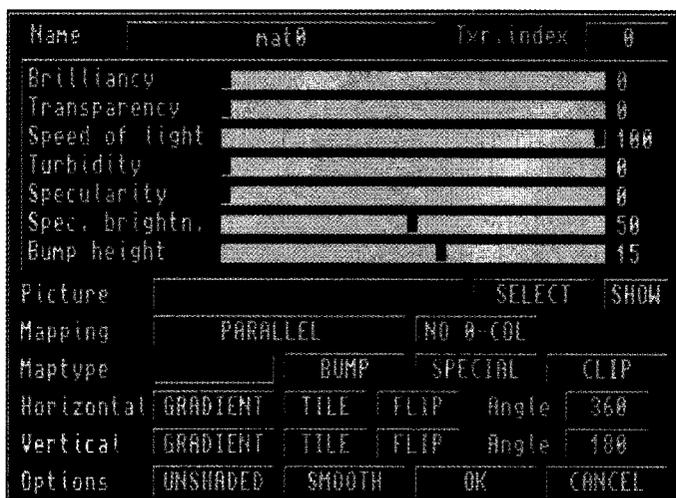
Questa operazione si rileva spesso molto utile, poiché le curve di base vengono create in modo un po' grezzo; in tal caso si potrà aumentare il numero dei punti, così da ottenere una figura più precisa. Al contrario, si può ridurre tale numero in una curva complessa rendendola così più facile da editare (un numero minore di punti si controlla meglio), per poi, alla fine, riportarla al suo grado di precisione iniziale.

Un altro vantaggio derivante dalla descrizione matematica dei solidi, è la già citata possibilità di eseguire operazioni Booleane (DIVIDE, AND, OR, AND NOT e OR esclusivo) tra due o più oggetti, una tecnica molto usata per la costruzione di figure geometriche e di parti meccaniche.

Combinati appositamente, gli operatori logici offrono uno dei mezzi più potenti per creare e modificare forme solide: il loro campo di applicazione è pressoché illimitato poiché gli oggetti utilizzati possono essere anche estremamente complessi o contenere risultati di precedenti operazioni Booleane. Inoltre, REAL3D è attualmente l'unico programma per Amiga che permetta di eseguire le operazioni logiche in modo che non abbiano effetto solo sul volume degli oggetti, ma anche sui materiali e sulle proprietà che caratterizzano le superfici interessate; questa è una delle caratteristiche più utili e innovative di questo software.

Ad esempio, sarà facile modellare una bellissima fetta di formaggio olandese (vedi figura 3), tagliando una sfera rossa con un piano giallo, e magari completando l'opera "scavando" con l'operatore AND NOT qualche buco sulla superficie ottenuta.

Degni di nota sono, infine, alcuni operatori, come i "Pixel Tools" e gli "Special Tools", che permettono di ottenere forme piuttosto inusuali in modo rapido e con semplicità.

Purtroppo però, malgrado la sofisticata tecnica di modellazione e i potenti operatori messi a disposizione, l'editor tridimensionale di REAL 3D fa sentire una generale carenza di precisione, qualora la scena che si vuole costruire non

Il Material Requester in configurazione di Default.

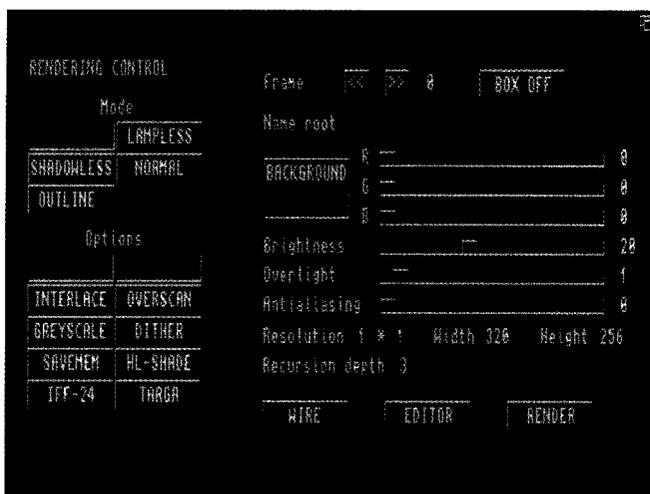
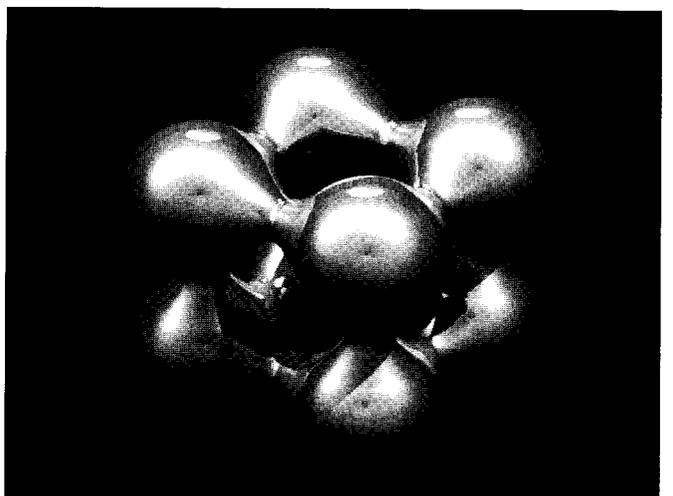
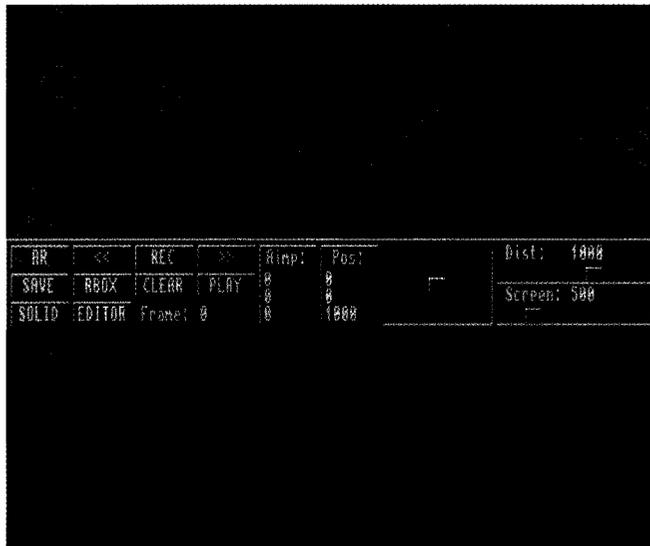
sia composta unicamente da primitive o oggetti ottenuti per rotazione; i problemi sono dovuti soprattutto alla mancanza di una proiezione assonometrica, e ad alcuni particolari non del tutto secondari, quali il quantomeno obsoleto metodo di posizionamento nelle tre dimensioni (lo stesso di Sculpt...) e una visualizzazione spesso sommaria degli oggetti sui quali si sta lavorando (basti pensare che il risultato di una operazione logica non è visibile nella "tri-view" ne, tantomeno, nel modo "wireframe"!).

LE PROPRIETA' DEI MATERIALI

Anche se integrato nell'editor tridimensionale, un discorso a parte merita il requester di definizione delle proprietà dei materiali di cui sono composti gli oggetti presenti in un'immagine.
Quest'ultimo, infatti, rappresenta sicuramente quello che si può definire il cuore del programma, poiché il risultato, ma soprattutto il realismo del lavoro finito, dipenderà direttamente dal lavoro eseguito in sede di definizione dei parametri fisici relativi alle superfici utilizzate.

Ancora un esempio di Rendering in Hi-Res a 16 colori.

REAL 3D in questo senso offre certamente una vasta gamma di possibilità, nonché una buona dose di flessibilità; è possibile, infatti, creare e modificare un numero pressoché illimitato di descrizioni di materiali, che potranno poi essere salvate in più librerie o in un unico file, permettendone così un futuro riutilizzo.
Tali descrizioni riguardano principalmente una serie di parametri fisici che determinano le interazioni esistenti tra le superfici e le luci presenti nella scena; il "Material Requester", infatti, permette di agire su variabili come la riflessività (quantità di luce riflessa dalla superficie di un dato materiale), la trasparenza, l'indice di rifrazione (strettamente correlato alla trasparenza), la densità e la specularità (o lucidità) di una data superficie.
In tal modo è possibile definire tutte quelle caratteristiche, colore a parte, che conferiscono all'oggetto non solo l'aspetto esteriore desiderato, ma anche alcune proprietà fisiche: è infatti possibile ottenere una vera lente di ingrandimento agendo sull'indice di rifrazione, oppure materiali particolari, come la nebbia e il fuoco (vedi la candela nella Figura 1), diminuendo appositamente il parametro di densità.
Tuttavia la parte più importante nella descrizione dei materiali è sicuramente la sezione relativa al cosiddetto "Brush Mapping", una tecnica utilizzata ormai in ogni programma di ray tracing che si rispetti e che, a mio avviso, rappresenta uno dei punti di forza di REAL 3D; con essa, attraverso la combinazione di varie opzioni, è possibile ottenere effetti che vanno oltre la semplice colorazione dell'oggetto; infatti, sotto tale nome, vengono raggruppati tutti quegli operatori che sono in grado di modificare il colore o uno qualunque dei parametri fisici sopradescritti, partendo dalla descrizione (in standard IFF) di una normale immagine bidimensionale. Accanto al normale "Color Mapping" (i colori del-

Lo schermo di settaggio dei parametri di Rendering.

Un esempio delle capacità di Real 3D.

l'oggetto derivano direttamente da quelli presenti nell'immagine utilizzata), figuranoaltri metodi di "mappatura" di brush, quali il "Bump Mapping" (col quale si possono ottenere bassorilievi e disturbi sulla superficie di un oggetto), lo "Special Mapping" (che permette di agire sulla trasparenza e sulla riflessività di una superficie soltanto in determinate aree e non sulla totalità dell'oggetto) e il "Clip Mapping".
Quest'ultima tecnica è sicuramente la più potente e la più innovativa visto che, combinata con un tool di eliminazione del colore nero ("No O-color"), è in grado evitare all'utente l'uso degli operatori Booleani; essa infatti, utilizzando una immagine a sfondo nero, lascerà visibili soltanto le parti colorate dell'oggetto, rimovendo le altre dall'immagine finale.
Tuttavia anche nel caso del "Brush mapping" vengono a galla alcune lacune, soprattutto in fase di posizionamento e dimensionamento delle brush; inoltre la possibilità di "mappare" una sola immagine per volta, può creare spesso qualche difficoltà di troppo.

Pannello di controllo del modulo Wireframe.

LE ANIMAZIONI

Ed eccoci giunti all'argomento che è croce e delizia di tutti gli appassionati della computer grafica: l'animazione.
E' croce non solo per le interminabili ore di calcolo che losviluppodi una anche breve animazione richiede, ma soprattutto per la complessità che in genere deve essere dominata, poiché, più saranno le opzioni disponibili, tanto più sarà complicata la gestione delle stesse.
E' invece delizia per lagrande soddisfazione che può dare una animazione ben riuscita, una volta terminato il lavoro.
Questo avviene, direi, per tutti i software di animazione tridimensionale e REAL 3D non fa eccezione.
Sicuramente il programma non si colloca al top della categoria per quanto riguarda le tecniche messe a disposizione dell'utente, tuttavia, nel caso specifico dell'animazione, è in grado di offrire un buon grado di flessibilità insieme a una discreta facilità d'uso.
Il manuale, sull'argomento, è come al solito molto sintetico, ma abbastanza esauriente.
Quattro sono le tecniche di animazione supportate da REAL3D.
La prima è basata quasi totalmente sul modulo wireframe, che permette il posizionamento dell'osservatore nella scena tramite movimenti e rotazioni eseguiti in tempo reale; i diversi frame vengono creati semplicemente "registrando" ogni variazione che l'utente imprime alla posizione o all'allineamento della telecamera.
Questo procedimento è sicuramente molto intuitivo, ma non offre grandi possibilità.
Al contrario, la seconda tecnica è un po' più complessa, ma permette di far uso di traiettorie (o path) definite nello spazio.
Questo metodo è sicuramente uno dei più usati per la definizione di movimenti nelle tre dimensioni, poiché è adatto alla quasi totalità delle situazioni che si possono presentare.
Modificando la lunghezza delle singole traiettorie è, inoltre, possibile gestire accelerazioni e decelerazioni con una tecnica che ricorda molto da vicino quella di Sculpt Animate 4D.
Durante il loro "viaggio" lungo i path creati dall'utente, è anche possibile far ruotare ogni singolo oggetto nelle diverse direzioni e per angoli arbitrari (sul manuale questa è la terza tecnica).
Infine, l'ultima tecnica, ma sicuramente la più potente a nostra disposizione, è in grado, oltre che ad abbozzare un principio di morphing (metamorfosi) tridimensionale, di eseguire operazioni di qualunque natura, come spostamenti, ridimensionamenti, rotazioni, su un qualsiasi oggetto, ridistribuendo l'effetto di tali operazioni su un dato numero di frame.
Indubbiamentequestoè un procedimento che offre un'enorme flessibilità, ma ciò va a discapito della semplicità d'uso; infatti, definire una macro com-

prendente un buon numero di passi, è paragonabile allo scrivere un piccolo programma in BASIC.
L'unico appunto comunque che mi sento di fare a questa sezione di REAL 3D, è la totale mancanza di un seppur rudimentale algoritmo di morphing tra oggetti, tecnica ormai utilizzata da quasi tutti i software di pari prestazioni.

IL RENDERING

REAL 3D include quattro differenti modi di rendering.
Il "Fast Mode" è un metodo molto rapido (l'algoritmo usato è probabilmente quello di Gouraud) che, convertiti tutti gli oggetti in un materiale di base non riflettente, non trasparente e privo di brush, ne visualizza il modello utilizzando una sola sorgente di luce posta nello stesso punto dell'osservatore; questo metodo si rivela utile in fase di preview dell'immagine e per la visualizzazione di oggetti la cui importanza (come nelle applicazioni di CAD) risiede nella forma, non nello shading.
In modo "Normal" viene utilizzato il modello completo di rendering, vale a dire che il risultato finale comprenderà ombre, riflessioni, trasparenze, ecc.
Il "Shadowless Mode" è sostanzialmente uguale al precedente, ma con un'unica eccezione: le ombre non vengono calcolate. Questo artificio riduce di parecchio i tempi di rendering, rendendo questo modo particolarmente indicato per la produzione di animazioni. Vi è, per ultimo, il "Lampless Mode", una combinazione dei due modi "Fast" e "Shadowless", che permette di abbattere ulteriormente i tempi di calcolo: viene utilizzata soltanto una sorgente di luce, ma le brush e le proprietà dei materiali vengono mantenute. Inoltre, è possibile calcolare solo una parte dell'immagine o ingrandire i pixel a piacimento, due funzioni molto utili durante le prove, che suoneranno familiari agli utenti del vecchio Turbo Silver. Bisogna, infine, segnalare che REAL 3D Professional supporta attualmente due tipi di frame buffer: l'Harlequin e l'austriaca VD2001 (poco conosciuto in Italia).

Figura 3: fetta di formaggio.

CONCLUSIONI

Credo che la valutazione globale di questo pacchetto non possa prescindere dall'uso che se ne vuole fare o dal prezzo (la versione Professional costa circa mezzo milione, mentre quella normale si aggira sulle 220 mila lire). Ne consiglierei l'acquisto a tutti coloro che desiderano produrre animazioni e immagini ad un buon livello, senza però addentrarsi troppo nei meandri di quel affascinante ma complesso mondo che è la grafica tridimensionale. Con REAL 3D infatti, fin tanto che si resta nell'ordine di idee delle classiche "scene da computer graphics" (cioè quelle composte principalmente da sfere ultralucenti, cubi riflettenti, scritte in movimento, scacchi e vasellame di ogni genere o forma), è possibile creare animazioni e immagini di sicuro effetto; tuttavia se si volesse spingersi un poco oltre e affrontare magari un progetto un po' più complesso, soprattutto per quanto riguarda la modellazione, comincerebbero a farsi sentire i limiti presenti soprattutto in fase di costruzione degli oggetti. In questi casi, oltre che vivamente raccomandato, è praticamente necessario l'utilizzo di altri pacchetti "a poligoni", forse un po' meno veloci e intuitivi, ma sicuramente più flessibili e potenti. A metà del 1992, secondo me, non si può più pensare di produrre un pacchetto di rendering e animazione 3D, prendendo come punto di riferimento un software obsoleto come Sculpt Animate (il manuale indugia spesso su palesi confronti con il programma della Byte by Byte); le nuove frontiere oggi sono rappresentate da Imagine e da Lightwawe, ed è a questi due concorrenti che la prossima release di REAL 3D dovrà ispirarsi. Quindi forza fratelli Meskanen! La strada imboccata è sicuramente quella giusta, ma ci vuole ancora un piccolo sforzo per emergere definitivamente.

SCHEDA PRODOTTO

Nome Prodotto: Real 3D Professional 1.4 - Turbo
Casa Prodruttrice: Realsoft Ky
Distribuito da: Activa International
In vendita: Applied Peripherals & Software - Corno di Rosazzo (UD) - Tel. 0432/759264
Prezzo: Lire 590.000 (vers. Pro) - Lire 220.000 (Beginner)
Giudizio: buono
Configurazione minima richiesta: Amiga con 1 Mb di RAM
Pro: semplicità d'uso, tempi di rendering, descrizione delle superfici
Contro: versatilità, poche brush, visualizzazione degli oggetti
Configurazione della prova: A2000b 1.3, 2.0, 5 Mb RAM

RAPPRESENTAZIONE PARAMETRICA DEI SOLIDI (SUPERFICI CURVE)

I poligoni sono sicuramente le primitive di modeliazione più usate per la loro semplicità d'uso e la loro versatilità. Combinandoli è possibile costruire o approssimare praticamente ogni tipo di oggetto. Si deve tener presente però che per modellare forme curve (una sfera ne è l'esempio più facile) utilizzando soltanto poligoni, si è costretti a ricorrere a interpolazioni del colore o della normale al punto considerato durante la fase di rendering. Questo approccio normalmente funziona, tuttavia vi sono casi in cui sorgono alcune controindicazioni. Infatti, visto che i calcoli effettuati influenzano soltanto il colore finale di un dato punto e non la forma dell'oggetto stesso, la sua silhouette resterà poligonale cosicché, se quest'ultimo verrà posto a breve distanza dall'osservatore, tale imprecisione potrà diventare talmente visibile da rovinare l'illusione creata dalle interpolazioni.
Inoltre, un ulteriore problema deriva dalle intersezioni tra oggetti poiché, lungo la loro linea di giunzione, la struttura poligonale sottostante diventerà evidente, rovinando ancora una volta l'illusione ottica. La rappresentazione parametrica dei solidi risolve questo tipo di problemi, poiché le superfici risultanti sono curve per definizione. Di tali superfici esistono parecchi tipi diversi e, anche se qualche volta risultano più difficili da gestire, in parecchie situazioni si rivelano più appropriate da usare rispetto ai più comuni poligoni. Tecniche che furono in origine sviluppate per modellare automobili e fusoliere di aeroplani in ambiente CAD, vengono ora utilizzate in molti settori differenti della computer grafica. Una delle rappresentazioni parametriche più usate è sicuramente la cosiddetta "superficie di Bezier", sviluppata appunto da Pierre Bezier verso la fine degli anni '60 per la Renault Il suo sistema CAD, in uso fin dal 1972, è senza dubbio il fautore di molti dei numerosi modelli di automobile prodotti della casa francese durante gli anni '70 Probabilmente il lavoro di Bezier era stato preceduto dai risultati ottenuti da P. de Casteljau alla Citroen nei primissimi anni '60, i suoi resoconti, però, rimasero nascosti in qualche archivio fino al 1975 Le rappresentazioni parametriche normalmente usate in computer grafica sono definite da una o più espressioni matematiche, piuttosto che da insiemi di coordinate nello spazio tridimensionale, garantendo così che anche i contorni dei solidi descritti siano realmente curvi In questo modo sarà possibile calcolare il colore e l'ombreggiatura di un punto qualunque utilizzando direttamente la normale al punto stesso e non semplicemente una sua approssimazione Vediamo ora, un po' più da vicino, i due tipi di rappresentazione parametrica che più si sono dimostrati utili agli scopi della computer grafica. Entrambi richiedono alcune sofisticazioni matematiche che non tratterò in questa sede; tuttavia, se qualcuno volesse farsi un'idea di come tali primitive funzionino, i miglior riferimento che posso dare è il libro "Computer Graphics, Prnciples and Practice di J. Foley e A. Van Dam", volume considerato a tutt'oggi la "Bibbia della computer grafica"

I PATCH

Una grande varietà di superfici curve aperte può essere raggruppata sotto il nome di "patch".
Il concetto che sta alla base di questa categoria di curve è il fatto che l'utente possa definire delle piccole sezioni (i patch, appunto) che, accostate l'una all'altra, formano superfici più grandi con le quali vengono poi costruiti oggetti anche molto complessi.
La definizione di patch è molto generale e comprende molti tipi di forme differenti; del resto anche i poligoni potrebbero essere considerati una primitiva appartenente alla classe dei patch.
Nella pratica, però, col termine patch si tende a indicare soltanto quelle primitive grafiche che presentano un minimo di curvatura propria, ottenuta in generale utilizzando curve di Bezier o B-spline. Alcuni sistemi permettono all'utente di specificare una struttura poligonale che verrà poi utilizzato come punto di partenza per la costruzione di un modello dell'oggetto costituito unicamente da patch. L'idea è quella di considerare la rappresentazione poligonale solo come un'approssimazione della superficie curva che il computer calcolerà in seguito; per modificare quest'utima, quindi, basterà agire sui vertici che compongono il modello poligonale da cui deriva. La costruzione dei patch, che avviene poligono per poligono, può essere di due diversi tipi: se la superficie curva passa da tutti i punti di controllo del poligono sottostante, essa prenderà il nome di "patch interpolante"; si parlerà invece di "patch approssimante", qualora essa si limiti a passare nelle vicinanze dei vertici del modello poligonale. Comunque il meccanismo che sta alla base di questo tipo di rappresentazione varia parecchio da sistema a sistema, anche perché esistono numerosi tipi di patch diversi, che possono inoltre essere usati in molti modi differenti.

LE SUPERFICI QUADRATICHE

Una superficie quadratica è formalmente definita come "quella superficie che è soluzione di un'equazione di secondo grado", tipicamente sono quadratiche superfici come la sfera, i cilindro, il cono, il paraboloide o l'iperboloide. Per questo tipo di superfici la rappresentazione in memoria è molto differente da quella utilizzata per poligoni o per i patch (è meno complessa e occupa una quantità molto minore di spazio in memoria per i dati). Una sfera, ad esempio, resta completamente definita una volta che si siano assegnati un centro e la lunghezza del raggio. Un cilindro necessita invece di un'asse centrale (che ne definisce l'orientamento), di un raggio (che ne determina lo spessore) e dei limiti in lunghezza agli estremi dell'oggetto; questi ultimi sono resi necessari dal fatto che a definizione matematica di molte superfici quadratiche è tale da far si che, in origine, esse abbiano un'estensione infinita, creando così un effetto che non sempre è desiderato. Benché non siano molto flessibili. le superfici quadratiche vengono utilizzate sempre più frequentemente, anche in alcuni modellatori per poligoni (la primitiva "sphere" di Imagine è una quadratica), in virtù della loro semplicissima descrizione che permette di ridurre drasticamente l'occupazione di memoria e i tempi di rendering E' possibile, comunque, aumentare la versatilità di questo tipo di curve innalzando il grado del polinomio che e descrive (per una curva terzo grado ad esempio, si parlerà di superficie cubica o bicubica); tuttavia non mi addentrerò nella descrizione di tali rappresentazioni, anche perché la loro applicazione è, per ora, quasi esclusivamente limitata al settore CAD.

Marco Pugliese

ANIMAKER

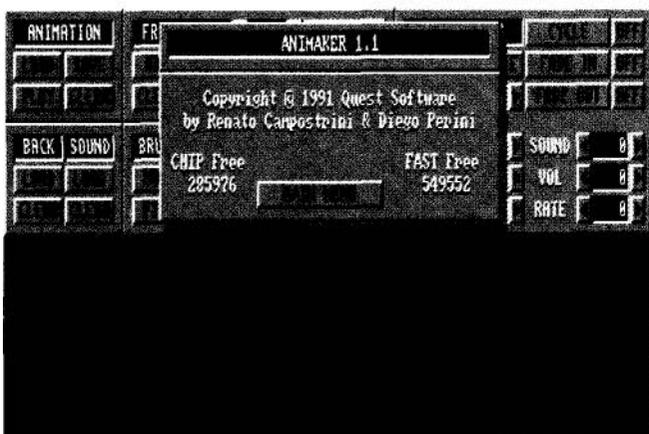
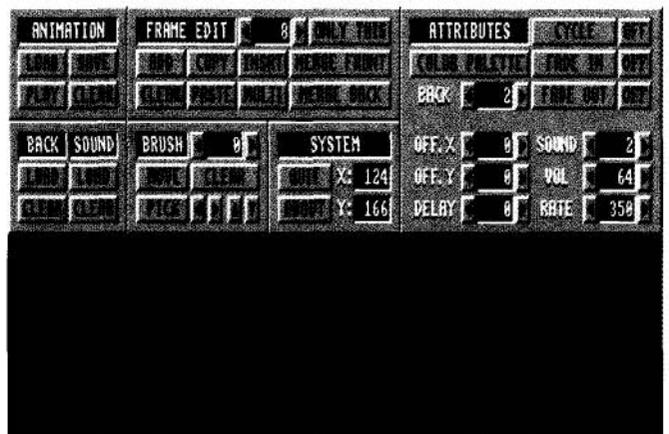
Stefano Paganini

Un ottimo programma di animazione.

Tra i parametri principali per la valutazione di uno slideshow, al di là della creazione delle singole immagini, vi è sicuramente il tipo di effetti speciali di contorno, l'assenza di errori nella gestione dei colori, durante le dissolvenze incrociate, e una buona sincronizzazione con l'audio.
I programmi di questo tipo, generalmente, permettono di porre, secondo una certa sequenza, immagini e suoni, con effetti speciali all'inizio e alla fine di ogni parte chiave dell'animazione.
Tra i molti programmi disponibili per Amiga, dai più semplici PAGEFLIPPER ai veri e propri sistemi di Authoring, vi è una gamma di scelta assai ampia.
Il prodotto in prova in questo articolo si presenta con una notevole dotazione di caratteristiche tecniche e in più due note degne di menzione, è stato sviluppato dalla Quest di Verona, già nota a molti utenti Amiga per la sua attività in questo settore, e ha un costo al pubblico inferiore alle 100.000 Lire.

UN'OCCHIATA AL PRODOTTO

AniMaker viene fornito in un solo dischetto, all'interno del quale si trovano, sia il programma vero e proprio, sia il player, sono inoltre contenuti due file .ANIM dimostrativi, uno per Amiga con un solo Megabyte di RAM, che risulta privo della traccia audio, e una versione completa dell'animazione, con audio, che mostra una panoramica delle funzioni disponibili tramite il programma.
Dato che il programma è italiano sembra strano che lo slideshow dimostrativo sia completamente in inglese: d'accordo per la prospettiva del mercato straniero, ma...
Un file AM-News include le novità incluse nella versione corrente 1.1 e specifica che le animazioni possono in realtà essere PLAY e solo con "almeno" 1,5 Mb di RAM, ferme restando le considerazioni di cui sopra, nel caso di animazioni semplici.
Il manuale in dotazione è spartano, essenziale e racchiude tutte le informazioni: la suddivisione è quella classica tutorial-guida di riferimento, anche se per un programma di grafica ci si sarebbe aspettati forse una maggior presenza di schermate.
In realtà tutte le funzioni, disponibili tramite un'interfaccia utente totalmente a bottoni, sono spiegate solo in modo testuale.
Ciò può risultare controproducente nei confronti degli utenti meno esperti per i quali, particolarmente la parte di tutorial, deve essere più curata.
Come specificato nel manuale, AniMaker può girare su un qualunque modello di Amiga.

Informazioni del programma

Il pannello di controllo principale.

In particolare gli utenti del 500 inespanso potranno beneficiare di questo programma, creando sequenze di buona qualità anche se limitate nel numero di schermate e nel sonoro. La regola della disponibilità di memoria, che vale per tutto il software per Amiga, e che si può tradurre nell'affermazione "mai abbastanza", è validaanche per AniMaker. Quindi al crescere della complessità dell'animazione, e soprattutto nel caso dell'utilizzo della traccia audio e della presenza di hard disk, i requisiti di memoria crescono prevedibilmente.

La configurazione minima consigliata per lavorare prevede almeno 3 Mb di RAM, tenendo conto della limitazione, tuttora vigente per buona parte degli Amiga installati, dei 512 K di Chip RAM.

Questo tipo di problemi risultano del tutto irrilevanti per gli utenti di Amiga 3000. E' da notare che, al contrario di quanto specificato sul manuale, la versione per il S.O. 2.0 non è presente sul dischetto e che quindi la configurazione minima prevede anche il vecchio 1.2.

Il programma è comunque sufficientemente ottimizzato, sia nell'occupazione della memoria, sia nella gestione di errori e situazioni critiche, primatratutte la segnalazione di "memoria insufficiente" che impedisce di creare animazioni con double-buffering, consentendo tuttavia di proseguire con animazioni più semplici.

La tecnica del double-buffering permette, tramite l'utilizzo di due pagine grafiche contemporaneamente, di visualizzare e caricare le immagini alternativamente, eliminando in questo modo lo sfarfallio tipico.

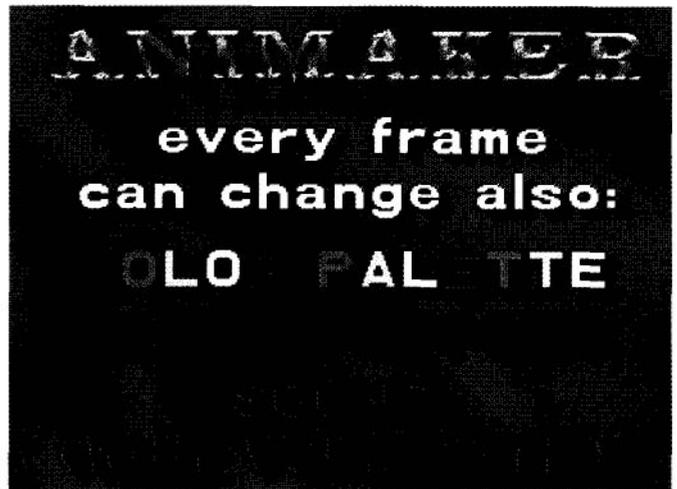
AniMaker puòessere installato anche su hard disk: non vi è una procedura particolare da attivare, basta spostare le icone dal dischetto sull'icona dell'unità DHx.

Il demo incluso in AniMaker.

ANIMAKER: IL SOFWARE

La schermata iniziale del programma presenta il pannello di controllo suddiviso in diverse zone, ciascuna delle quali contiene alcuni bottoni e campi per la definizione dei parametri.

Alcuni bottoni attivano a loro volta altri pannelli di selezione, come, ad esempio, per il caricamento di file esterni.

La suddivisione in zone può essere riassunta nel seguente modo:

- Animation
- Frame edit
- Attributes
- System
- Brush
- Back & Sound

Fondamentalmente la creazione di un'animazione con AniMaker si svolge in questo modo: vengono create, tramite i ben noti programmi di paint e rendering, le schermate di sfondo (background) e i brush, che potranno essere mossi come oggetti per tutto lo schermo (il formato IFF è supportato così come tutte le modalità grafiche di Amiga comprese HAM e EHB).

Allo stesso modo, tramite programmi di sound editing, vengono selezionate le parti che compongono la colonna sonora: anche in questo caso il formato è lo standard IFF 8SVX o RAW.

Successivamente, all'interno di AniMaker, tutte le parti vengono caricate e messe nella giusta sequenza.

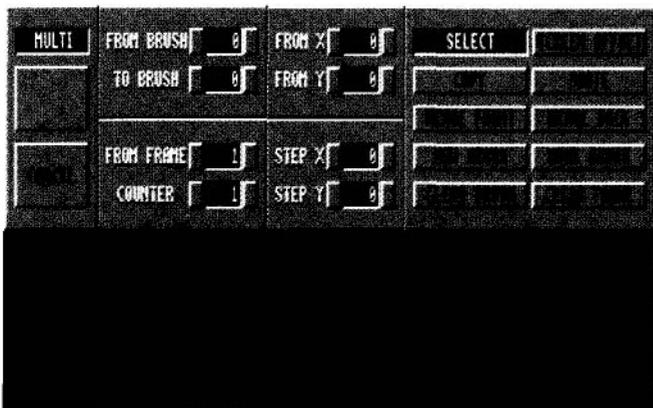
Soffermiamoci su questo punto, AniMaker supporta un numero massimo di 9999 fotogrammi, ed è questa "l'unità di misura" del programma, cioè ogni fotogramma ha al suo interno diverse "azioni" possibili: spostamentoltraslazione sia di un oggetto/brush sia del background, con la definizione dello spostamento in pixel.

L'operazione viene quindi condotta fotogramma per fotogramma, anche se con la possibilità di far "ereditare" alcune modifiche ad oggetti e caratteristiche anche ai fotogrammi successivi (bottone ALSO NEXT/ONLY THIS).

Ogni frame può avere a disposizione un certo numero di sfondi, al massimo 100, altrettanti oggetti/ brush, da modificare in posizione assoluta ed eventualmente sovrapporre ad altri brush.

Anche il suono può essere modificato per ogni fotogramma, in frequenza, volume e durata.

Lasceltadi questa impostazione rende da un lato più semplice e veloce la creazione di sequenze rapide e racchiude in sé un indubbio vantaggio nell'editing di ogni singoloframe.

Tuttavia, quando l'animazione cominciaad essere di un certo peso (numero fotogrammi), si nota la difficoltà

Il pannello Multi.

nell'editing veloce e, più in generale, la mancanza di uno storyboard che racchiuda in forma simbolica e sintetica tutte le operazioni per ogni frame.
Un'altra caratteristica che manca è quella della definizione di key frame, cioè di sequenze chiave, all'interno delle quali vengono definite delle azioni, che vengono poi sviluppate automaticamente in un certo numero di frame successivi.
Durante la prova abbiamo potuto constatare che la mancanza di queste funzioni può essere, in parte, compensata da una certa esperienza nell'utilizzo dello stesso AniMaker.
Uno degli esempi più evidenti è dato dai classici effetti di dissolvenza che possono essere utilizzati solamente per un frame, oppure estesi a più frame, includendo anche cambiamenti di palette, sonoro o altro elemento: questa caratteristica trae vantaggio dall'organizzazione dei frame solo se opportunamente sfruttata.
Complessivamente il programma permette di sincronizzare sia audio che video con un buon margine di precisione, inoltre l'unità di tempo che determina il passaggio da un frame all'altro (che è di 1/60 sec.) è indipendente da quella della parte sonora che può così essere modificata in piu riprese, in più frame.
L'animazione che ne risulta viene salvata in un file .ANIM che però non segue lo standard ANIM ben noto, ma che differisce, oltre che per le specifiche, anche per l'inclusione di alcune caratteristiche fondamentali quale, ad esempio, la possibilità di cambiare modalità grafica, o anche la sola palette, da un frame all'altro.
Anche questa scelta, se da un lato permette di compattare tutte le componenti dell'animazione in un unico blocco, sempre nel caso di una animazione "notevole", può risentire della mancanza di spazio su dischetto: a tale proposito è da citare il fatto che il demo completo, che dura qualche minuto e con due sole tracce audio, occupa 408 K, e nella versione ridotta 267 K (cioè senza audio).
Inoltre, è un vero peccato che non si possano includere sequenze 3D, proprio in formato ANIM, come background: tale caratteristica avrebbe reso il programma veramente completo e, sebbene comporti una complessità non indifferente in fase di programmazione, ci auguriamo sia inclusa in una versione futura di AniMaker.

IL PANNELLO MULTI

Alcune delle caratteristiche avanzate del programma vengono attivate tramite il pannello Multi. Il pannello Multi permette di effettuare modifiche globali sui frame e sui brush dell'animazione. Ciò significa che ogni bottone di questo pannello consente di modificare parametri di schermo e/o di brush, specificando quali frame ne siano coinvolti.
Tra le operazioni possibili a livello globale, oltre a quelle già disponibili a livello frame, vi sono quelle di TAGLIA/INCOLLA che, facendo uso di una zona di memoria temporanea, permettono di applicare determinate modifiche ad un gruppo di frame, mentre per i brush è possibile aggiungerne di nuovi e specificarne posizione e step incrementale per lo spostamento.

CONCLUSIONI

La prova e le relative valutazioni non possono non tenere conto di alcuni fattori:

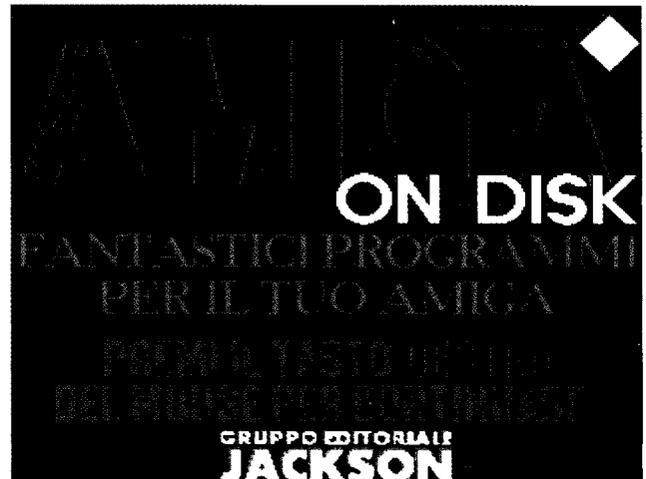
- il prodotto costa al pubblico 89.000 Lire
- si tratta di un prodotto italiano
- è la versione 1.1

Il programma così com'è è in grado di lavorare egregiamente per piccole animazioni, veloci e con simpatici effetti speciali, senza contare la parte audio.Tuttavia sussistono dei problemi per la creazione di animazioni complesse. ▲

SCHEDA PRODOTTO

Nome Prodotto: AniMaker 1.1
Casa Prodruttrice: Quest - Verona
Distribuito da: Soundware - Varese - Tel. 0332/232670
Prezzo: Lire 89.000
Giudizio: buono
Configurazione richiesta: Amiga con 3 Mb (consigliati)
Pro: rapido, versatile e prezzo basso
Contro: struttura poco flessibile per applicazioni pesanti
Configurazione della prova: A500 512K, 1 drive.

Scopri i segreti di AMIGA

ON DISK è una rubrica mensile di quattro pagine che possono anche essere staccate e conservate; in queste pagine sono descritte tutte le informazioni dei programmi inclusi nel disco, complete di istruzioni, trucchi ecc... In questo spazio troveranno posto giochi, utility e tutto ciò che può fare Amiga.

• Games

Space War v1.11

Jeff Petkau/Brian Fehdrau

Space War è stato il primo videogioco spaziale della storia, inventato nel 1962 da un gruppo di studenti del MIT in America Realizzato inizialmente su un mainframe molto costoso, utilizzava una grafica vettoriale (non era ancora possibile realizzare della grafica bitmap) e permetteva a due giocatori di giocare contemporaneamente uno contro l'altro Molto tempo è passato da allora, l'industria dei videogiochi si è evoluta in maniera straordinaria e voi potete giocare sul vostro Amiga a Space War, ORA Non è meraviglioso tutto questo7 A Space War si gioca ovviamente uno contro l'altro anche su Amiga e questo comporta necessariamente che due giocatori "umani" prendano parte alla sfida Lo scopo del gioco è semplicemente distruggere l'astronave nemlca, ma molti fattori esterni entrano "in gioco" Al centro dello schermo è presente un sole dotato di una forza di attrazione gravitazionale e un asteroide gli gira costantemente intorno e rischia di urtare i giocatori Le astronavi dispongono di una forza di spinta che permette loro di muoversi nello spazio, evitare gli ostacoli e contrastare la forza di gravità del sole Tutti i parametri possono essere modificati a piacimento dall'utente: può essere rimosso il sole dal centro dello schermo o sostituito con un buco nero, può essere variata la forza di gravità e la potenza di spinta delle astronavi, si può eliminare l'asteroide e cambiare altri utili parametri di gioco. Alle opzioni principali si accede tramite il menu "Options", mentre il tipo di controllo per ogni astronave si sceglie dal menu "Control". Space War è, come abbiamo detto, un gioco esclusivamente a due giocatori, ma ciò non significa obbligatoriamente che i due giocatori debbano giocare con lo stesso computer con questa versione di Space War è addirittura possibile giocare in link con un computer collegato via modem A questo scopo è dedicato il menu "Remote" che fornisce tutte le opzioni per il controllo del modem e del computer che si trova dall'altra parte del telefono. Nella directory Space War sono comunque presenti dei sample di buona qualità e delle schermate grafiche che grazie allo standard IFF possono essere modificate, se lo si desidera, per ottenere una versione personalizzata del gioco. Space War oltre ad essere molto divertente come ogni gioco che si può giocare in due, ha anche un significato storico. Un mito da non sottovalutare!

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstari 1.2/1.3/2.0

UTILIZZO
Workbench: Doppio Click sull'icona

FILE DI SUPPORTO
Tutti i file presenti nella directory "Space-War" del disco AmigaMagazine

• Utility

MagicFileRequester

Stefan Stuntz

Questa è l'utility dedicata esclusivamente a chi dispone di KickStart 2.04 (Amiga 3000, Amiga 500 Plus). MagicFileRequester è stato progettato e realizzato come una commodity, non richiede quindi una libreria per funzionare, basta solo eseguirlo nella Startup-Sequence o riporre la sua icona nel cassetto WBStartup del vostro disco di lavoro. Una volta lanciato, MagicFileRequester si sostituisce al requester ASL standard sotto 2.0. Ciò che si ottiene è un file requester decisamente eccezionale, Sembra che il programmatore abbia avuto come unico scopo quello di realizzare il mi-

glior file requester della storia di Amiga, includendo delle comodità veramente interessanti. E' possibile, ad esempio, controllare tutte le funzioni del requester utilizzando esclusivamente la tastiera, è completamente configurabile tramite il menu "Information" del Workbench eseguito sulla sua icona, è possibile selezionare dei programmi che verranno utilizzati per ascoltare suoni, visualizzare immagini, ecc... ma soprattutto offre un aspetto di impaginazione grafica molto piacevole e funzionale, specie se viene utilizzato in alta risoluzione e/o con font proporzionali.
Attenzione: MagicFileRequester necessita di un gran numero di librerie per funzionare (l'elenco è riportato nelle "Specifiche del Programma") e quindi non è possibile eseguirlo cliccando sull'icona presente nel disco AmigaMagazine. E' necessario prima installarlo nel vostro sistema spostando la sua icona nel cassetto WBStartup del vostro disco WorkBench (usate una copia di lavoro, mi raccomando).

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K **RAM**
Kickstart 2.0

UTILIZZO
Workbench: Doppio Click sull'icona

FILE DI SUPPORTO
Le librerie intuition.library, graphics.library, gadtools.library, diskfont.library, icon.library, commodities.library, utility.library, asl.library e req.library DEVONO essere presenti nella directory LIBS del disco da cui si effettua il boot.

PC-Task

Chris Hames

E ora un programma che arriva direttamente dall'Australia. PC-Task è un emulatore PC, ossia un programma che rende il vostro Amiga completamente identico ad un computer della IBM, con la possibilità di eseguire tutto il software scrittto per quel tipo di computer In passato avete gia avuto la possibilita di vedere programmi simili a questo, come Transformer o IBeM, ma PC-Task li batte tutti in quanto a configurabilità e, cosa ben più importante, velocità Da test effettuati sugli emulatori disponibili, è risultato che PC-Task emula un PC circa due volte piu veloce rispetto ai suoi programmi "colleghi" su un Amiga dotato di un normale 68000 Ovviamente le prestazioni salgono con la velocità del microprocessore e con una scheda 68030 (o, magari, 68040) potreste rimanere sconvolti nell'apprendere che PC-Task riesce ad emulare un PC a più di 20-22 MHz Ad una velocità paragonabile a quella offerta da emulatori hardware come Janus AT 386 o ATOnce, ma ad un prezzo molto inferiore
Già, ad un prezzo inferiore, perché PC-Task è uno di quei programmi che si devono comprare per utilizzare al meglio delle sue possibilità
Chris Hames, il programmatore di programmi come DirWork, VirusMemKill, Degrader e molti altri, ha distribuito in Pubblico Dominio una versione dimostrativa di PC-Task che offre la possibilita di eseguire programmi In ambiente MS-DOS, ma con delle funzioni disabilitate In questa versione non e possibile scrivere verso alcun dispositivo di input/output, quindi non si può salvare nessun file su floppy o hard-disk e non si possono usare le porte parallela e seriale Per ottenere una versione completa e funzionante di PC-Task è necessario divenire utenti registrati mandando 40 dollari a Chris, e considerando la qualità di questo emulatore possiamo tranquillamente incoraggiare i lettorl a spendere 50 000 lire per acquistare un programma realmente professionale.
Ma torniamo alla demo di PC-Task Una volta cliccato sull'lcona, apparirà una schermata di configurazione in cui e possibile selezionare i colori dello schermo PC, i drive che verranno utilizzati, i parametri di priorità, ecc
Per fare una rapida prova basta semplicemente cliccare su start e inserire il disco di sistema MS-DOS nel drive L'emulazione PC dovrebbe inzrare vedendo il classico schermo nero con scritte grigie (sempreché non abbiate cambiato i colori di default) A questo punto siete a tutti gli effetti in ambiente MS-DOS, potete eseguire comandi, lanciare programmi sempre rimanendo nelle limitazioni di una versione dimostrativa Per uscire dall'emulazione è necessario premere contemporaneamente i due tasti Amiga congiuntamente al tasto Del, mentre per usufruire del multitasking che rimane attivo anche durante l'emulazione, basta premere i tasti Amiga-Sinistro+N

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K **RAM**
Kickstart 1.211.312.0

UTILIZZO
Workbench: Doppio Click sull'icona

FILE DI SUPPORTO
Se disponete di Kickstart 1.2 o 1.3 è necessario prima di cliccare sull'icona di PC-Task, cliccare sull'icona TDPatch 1213. E' necessario un disco di sistema MS-DOS su disco 720 K DSDD.

Roses

Carmen Artino

Anche questo mese un po' di relax Dopo tante utility più o meno impegnative. un programma che SEMBRA un programma "leggero", in realtà si rivela un modo affascinante per analizzare la parte della matematica che riguarda le equazioni polari. Ma di questo parleremo dopo aver dato uno sguardo al programma.
Per lanciare questo programma è sufficiente attenersi alle indicazioni riportate nella sezione "Specifiche del Programma". Una volta partito, Roses è controllabile tramite dei

Codici rilevatori d'errore a ridondanza ciclica

CRC a 16 bit e a 32 bit (seconda parte)

Alberto Geneletti

L'algoritmo di decodifica

L'ultima volta avevamo concluso l'esame dell'algoritmo di costruzione della codifica CRC, passiamo ora dalla parte di chi riceve e verifica la correttezza del messaggio. Sempre utilizzando l'analogia polinomiale pensiamo al messaggio ricevuto come alla somma del messaggio trasmesso T(X) e di un eventuale polinomio di errore E(X). In formule avremo:

$$T'(X) = T(X) + E(X)$$

Purtroppo il polinomio E(X) è incognito; va sottolineato, infatti, che la conoscenza di E(X) permetterebbe la ricostruzione del messaggio originale, poiché ogni 1 nella sequenza di bit di E(X) sarebbe indicativo del fatto che il corrispondente bit di T(X) abbia assunto un valore diverso in T'(X), e sarebbe quindi sufficiente invertirne lo stato. Si procede allora alla divisione tra T'(X) e il polinomio generatore G(X), che, come avevamo affermato nella prima parte, deve essere noto anche in ricezione (in pratica vengono adottati polinomi standard). Nel caso in cui questa divisione dia resto non nullo, saremo certi della presenza di un errore, poiché questa situazione sarebbe indicativa di un E(X) diverso da zero.

Tuttavia, nel caso E(X) risulti divisibile per G(X) non saremo ancora in grado di rivelare l'errore, in quanto anche T'(X), al pari di T(X), risulterà divisibile per G(X). Si tratta allora di scegliere un polinomio G(X) in grado di annullare statisticamente la probabilità che possa verificarsi un E(X) anch'esso divisibile per G(X). Per questo si parte dallo studio dei possibili E(X), che vengono ancora una volta considerati sotto forma di polinomi, e vengono forniti dei criteri algebrici che garantiscano la rilevazione dei vari tipi di errore modellizzati.
Consideriamo, ad esempio, l'errore singolo; ci troveremo allora davanti a un messaggio che presenta un solo bit di valore diverso rispetto all'originale. Detta K la posizione di tale bit nella sequenza originale, verrà prodotto un polinomio di errore del tipo:

$$E(X) = X^{K-1}$$

E' evidente che se G(X) è costituito da almeno due termini (fatto garantito dalle ipotesi, indipendentemente dalla scelta del polinomio), il resto della divisione tra E(X) e G(X) non potrà certamente essere nullo.

La codifica CRC è quindi in grado di rivelare implicitamente tutti gli errori singoli. Consideriamo ora il caso degli errori doppi, e precisamente siano N e M le posizioni dei due bit errati all'interno di T'(X), e sia N < M. Il polinomio di errore sarà allora del tipo:

$$E(X) = X^{N-1} + X^{M-1} = X^{M-1}(X^{N-M} + 1)$$

Dal momento che X alla M-1 non è sicuramente divisibile per G(X), per quanto è stato visto in precedenza, dovremo cercare un G(X) che non sia mai divisibile per un generico:

$$X^{K} + 1$$

una condizione soddisfatta anche per K molto alti da polinomi abbastanza semplici. I codici polinomiali permettono, inoltre, di rivelare in modo molto semplice un qualsiasi numero di errori dispari. In questo caso, infatti, il polinomio E(X) non risulterà mai divisibile per X + 1, poiché moltiplicando per X + 1 un qualsiasi polinomio, si otterrebbe un polinomio prodotto contenente un numero pari di termini, poiché nel caso si presentino due potenze di ugual grado tra gli addendi del polinomio prodotto, queste si eliminerebbero automaticamente (1+1 = 0). E' sufficiente allora inserire X + 1 tra i fattori di G(X).

Sono stati studiati criteri più complessi, sui quali non ci dilungheremo, che hanno portato alla formulazione di polinomi standard, omologati dall'ente internazionale per la definizione degli standard nel campo della telecomunicazione, CCITT, e che oltre a garantire la rilevazione dei tipi di errore sopraelencati (singoli, doppi e dispari), permettono di rilevare errori di tipo burst (i più comuni), errori che si verificano cioè su una sequenza di bit adiacenti.

Naturalmente, questi polinomi forniscono la certezza soltanto per messaggi di una limitata dimensione (1024 bit per i codici a 16 bit), dopodiché rimangono ancora validi, ma solo in termini probabilistici. Anche in questo caso studi

statistici hanno portato alla formulazione di criteri che minimizzano la possibilità che un errore possa passare inosservato; nello stesso tempo sono stati studiati polinomi generatori più lunghi (32 bit), in grado di fornire maggiori garanzie anche su messaggi di dimensioni elevate al prezzo di pochi bit di ridondanza in più.

Implementazione in linguaggio C

Nella directory Transaction del dischetto allegato, troverete tre sorgenti in linguaggio C che implementano, utilizzano ed esemplificano algoritmi per la codifica e la decodifica di messaggi con checksum CRC.
Il più importante, dal punto di vista teorico, è senza dubbio CRCUtils.c, dove vengono implementate alcune funzioni che possono essere richiamate da qualunque programma, dal momento che operano su un buffer di dimensioni arbitrarie.

In questo buffer deve essere memorizzato il messaggio da codificare (in termini algebrici il polinomio M(X)), del quale dovrà essere passata, come argomento a ciascuna funzione, la relativa lunghezza. Tuttavia, il messaggio non dovrà occupare completamente il buffer, ma dovranno essere previsti in coda al messaggio almeno due o quattro byte liberi, che verranno successivamente occupati dal checksum a 16 o 32 bit.

La prima funzione, DoCRC( ), calcola proprio il resto della divisione polinomiale tra M'(X), ottenuto da M(X) azzerando i byte extra in coda al buffer, e G(X), che, a seconda dei casi, occuperà una word o una longword, restituendo poi il valore del checksum sotto forma di numero intero a 32 bit, oltre a copiare il checksum stesso in coda al buffer.

La seconda, GetCRC( ), restituisce, sotto forma di intero, il valore del checksum posto in coda al buffer, e viene richiamato internamente dalla funzione successiva. Quest'ultima, CheckCRC( ), viene utilizzata in fase di verifica; una volta ricavato il checksum di coda, per mezzo di GetCRC( ), se ne verifica l'esatta corrispondenza con quello del messaggio, che viene ricalcolato; viene restituito poi un opportuno codice di controllo.

Queste tre funzioni sono state implementate tanto nella versione a 16, quanto in quella a 32 bit, utilizzando i due seguenti polinomi standard definiti dalla CCITT:

$$G16(X) = x^{16} + x^{12} + x^5 + 1$$

per i 16 bit e:

$$G32(X) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

per quanto riguarda i 32 bit. Questi stessi polinomi vengono utilizzati nei protocolli di comunicazione XModem e ZModem.

Dovendo utilizzare per la codifica di tali polinomi una word o una longword, per un totale di 16 o 32 bit, può sembrare strano che i gradi dei due polinomi siano rispettivamente 16 e 32, piuttosto che 15 e 31. La codifica di tali polinomi generatori richiederebbe, infatti, rispettivamente 17 e 33 bit (uno per ogni coefficiente di X alla n); la scelta del grado di questi due polinomi è stata effettuata tuttavia basandosi sulla pratica, che ha evidenziato la possibilità di costruire algoritmi di decodifica molto semplici (e di conseguenza molto veloci), nel caso il coefficiente di grado massimo, o bit più significativo, sia sottinteso e definito implicitamente a 1. Del resto si tratta di un'ipotesi già introdotta in precedenza per la fisica realizzabilità di circuiti logici di codifica e di decodifica, e non impone quindi limitazioni aggiuntive.

Nella nostra implementazione, abbiamo assegnato i coefficienti all'interno della word e della longword di definizione in sequenza da destra verso sinistra, assumendo che tale sia anche l'ordine dei bit che costituiscono il messaggio da codificare. Alcune implementazioni, come lo ZModem a 32 bit, utilizzano invece un G(X) rovesciato, poiché tale è l'ordine dei bit ricevuti dai dispositivi di telecomunicazione per i quali è stato progettato tale protocollo.

L'algoritmo utilizzato nella nostra implementazione calcola il checksum in modo iterativo, effettuando operazioni logiche tra ogni byte del messaggio e il checksum parziale. Per ogni byte del messaggio viene creata una maschera a 8 bit, che inizialmente presenta settato il bit più significativo; successivamente tale maschera viene shiftata per 8 volte verso destra, permettendo di testare, per mezzo di un and logico, ogni singolo bit del byte corrente.

Si ricava, quindi, lo stato del bit più significativo del checksum o resto parziale corrente; come abbiamo detto in precedenza, lo stato di tale bit deciderà se dovremo effettuare o meno l'OR esclusivo tra il messaggio e il resto parziale. Il checksum viene poi shiftato di una posizione verso sinistra; in questo modo il bit più significativo viene perso, mentre il meno significativo verrà rimpiazzato da quello del byte corrente definito nella maschera per mezzo di una OR condizionata. Essendo inizialmente il checksum completamente nullo (presentando quindi, di volta in volta, il bit più significativo a zero) l'effetto dei primi 16 o 32 cicli consisterà unicamente nella copia dei primi due o quattro byte del messaggio nel checksum. L'effetto dei cicli successivi può, invece, essere meglio compreso osservando la figura. A questo punto, infatti, il primo bit del checksum è pari a 1 (e precisamente all'1 in alto a sinistra della prima linea, corrispondente al dividendo stesso).

Dobbiamo allora sottrarre il prodotto del primo bit del quoziente corrente (che, come è possibile intuire, non viene nemmeno calcolato) per G(X), prodotto che in questo caso è pari a G(X), dal resto parziale corrente, il che equivale a effettuare l'OR esclusivo con G(X); occorre tuttavia ricordare che il nostro G(X) è in effetti privo del primo bit, corrispondente a X alla 16 o a X alla 32. Inoltre,

lo XOR coinvolge anche un nuovo bit del messaggio.

Tuttavia osserviamo che al passaggio successivo, il primo bit, che non viene rappresentato nemmeno in figura, varrebbe comunque zero, ed è quindi perfettamente inutile effettuare lo XOR su questo diciassettesimo bit. Shiftiamo allora il checksum di una posizione verso destra, introducendo un nuovo bit, ed effettuiamo l'OR esclusivo; nel passaggio successivo quest'ultima operazione logica verrà invece saltata, poiché il bit più significativo del nuovo checksum così ricavato è nullo. Questo accadrà in corrispondenza di ciascun bit nullo del quoziente. Terminati tutti i bit di ciascun byte si riprende con il byte successivo, reinizializzando la maschera; al termine dell'intero messaggio il checksum corrisponde proprio al valore cercato.

XModem e ZModem

Il protocollo di comunicazione XModem è presente nei pacchetti di comunicazione in diverse versioni. La più antica, chiamata semplicemente XModem, è completamente priva di codici di rilevazione di errore CRC, ma utilizza un semplice checksum tradizionale calcolato su blocchi di 128 byte; tuttavia l'inefficacia di tale codifica ha portato all'introduzione di nuovi protocolli, fra i quali l'XModem/CRC, che adottano il polinomio a 16 bit CCITT applicato su blocchi di dimensioni variabili da 256 a 1 k byte (XModem/1K). Nelle prime implementazioni questi protocolli utilizzavano un algoritmo di codifica simile a quello che abbiamo visto nel paragrafo precedente; in seguito sono stati messi a punto metodi tabellari, che permettono di velocizzare le operazioni di codifica e decodifica ricavando direttamente da una tabella il risultato degli shift a bit dell'algoritmo stesso, che costituiscono in effetti il punto critico del ciclo principale, poiché un'operazione di scorrimento comporta un tempo di esecuzione in cicli macchina mediamente dieci volte superiore a un'operazione logica.

Una strategia analoga è stata utilizzata nell'implementazione del protocollo ZModem, attualmente il più veloce e affidabile, che utilizza in modo intercambiabile tanto codici a 16, quanto codici a 32 bit. La caratteristica principale di questo protocollo, tuttavia, riguarda non tanto il grado del polinomio generatore utilizzato (che come abbiamo detto in precedenza è quello standard CCITT rovesciato), ma la lunghezza dei blocchi codificati.

```
1 0 1 1 0 1 1 0 0 0 0 | 1 1 0 0 1
----------------------+--------------
1 1 0 0 1             | 1 1 0 1 0 0 1
  1 1 1 1 1 1 0 0 0 0 |
  1 1 0 0 1           |
    0 1 1 0 1 0 0 0 0 |
    0 0 0 0 0         |
      1 1 0 1 0 0 0 0 |
      1 1 0 0 1       |
        0 0 1 1 0 0 0 |
        0 0 0 0 0     |
          0 1 1 0 0 0 |
          0 0 0 0 0   |
            1 1 0 0 0 |
            1 1 0 0 1 |

Dividendo :
1 0 1 1 0 1 1 0 0 0 0
Divisore :
1 1 0 0 1
Quoziente :
1 1 0 1 0 0 1
Resto :
0 0 0 1
```

Lo ZModem opera, infatti su blocchi di dimensioni variabili; inizialmente viene inviato un pacchetto di dati di dimensioni limitate; successivamente viene tentata la trasmissione di pacchetti sempre più lunghi, sempre con relativo checksum, fino a un limite di 8K. In seguito al verificarsi di un errore, tale dimensione viene nuovamente ridotta, fino al raggiungimento di una dimensione ottimale, che verrà nuovamente incrementata nel caso venga a mancare il rumore di disturbo sulla linea.

Metodi tabellari

L'ottimizzazione degli algoritmi di codifica e di decodifica per mezzo di metodi tabellari è nata da un'idea di Chuck Forsberg, della Omen Technology, lo stesso autore del protocollo ZModem. Forsberg ha implementato una macro che, chiamata iterativamente, fornisce checksum parziali CRC, ottenute a partire dal checksum precedente e dal nuovo byte, effettuando due sole operazioni di scorrimento. In pratica il byte più significativo del checksum viene utilizzato per indicizzare una tabella di 256 elementi a 16 bit, nella quale sono stati memorizzati opportuni valori, calcolati in modo da fornire il valore del nuovo checksum una volta effettuato l'OR esclusivo con il nuovo byte. Nel 1986 tale macro è stata migliorata ulteriormente da Stephen Satchell e, nella versione attuale, utilizza una tabella di valori calcolata da Mark G. Mendel, della Network Systems Corporation.

La versione a 32 bit rovesciata è invece opera di Gary S. Brown, che ha ottimizzato ulteriormente la codifica; per ottenere una corretta trasmissione dei bit del messaggio, prevista dal CRC a partire dal bit più significativo verso il meno significativo, Brown ha pensato di presentare al proprio DCE sequenze rovesciate, dal momento che quest'ultimo trasmetteva i bit sequenzialmente dal meno significativo al più significativo. La relativa tabella consta di 256 elementi a 32 bit, e come nel caso precedente, contiene valori corrispondenti allo shift e all'OR esclusivo tra tutte le combinazioni di checksum parziali. In questo caso l'indicizzazione della tabella avviene a partire dal byte meno significativo di un OR esclusivo tra il checksum e il nuovo byte.

I listati delle macro e delle tabelle di ottimizzazione possono essere trovati all'interno dei sorgenti delle librerie XPRXMO34, che implementa l'XModem/CRC e della XPRZ250, la versione più recente dello ZModem, che utilizza il CRC rovesciato a 32 bit, disponibili nelle aree destinate al software di comunicazione dalle migliori BBS.

Architettura dei computer Amiga

Un primo sguardo (prima parte)

Paolo Canali

Premesse

In questo articolo vedremo in dettaglio qual è la struttura dei personal Amiga, a partire dai concetti ispiratori, per finire con il modo in cui funziona il chip-set che è alla base delle loro notevoli capacità.

Non starò a ripetere le cose banali note a tutti coloro che hanno un po' di dimestichezza con microprocessori e computer: esistono moltissimi libri e articoli che spiegano che cosa sia una RAM, un bus o una ROM. Nella bibliografia ho indicato alcuni di questi libri, perlopiù (c'est la vie...) in inglese.

Non intendo però neppure fare un semplice riassunto dei due testi canonici per Amiga: l'A500-2000 Hardware Reference Manual e l'Hardware Manual. Cercate lì dentro ulteriori dettagli. Un po' di conoscenza della struttura hardware del 68000 può aiutare la comprensione dell'articolo, cercherò di illustrare i concetti essenziali a beneficio di chi non li conosce ancora.

Quanto vedremo si applica a tutti i modelli di Amiga prodotti finora; se occorrerà scendere a un livello di dettaglio maggiore, farò riferimento in genere agli Amiga 2000, 500 e 500+, essendo A3000 e A3000T molto più complessi e quindi poco adatti ad illustrare i concetti di base. Tenete a portata di mano gli schemi elettrici, che provvidenzialmente potete trovare in fondo al manuale del computer (il che è un fatto più unico che raro per i computer venduti in questi anni !).

Prima di proseguire, per dissipare eventuali perplessità, ricordo che l'A2000B e l'A2000A, pur avendo la stessa forma e (nominalmente) gli stessi slot, sono circuitalmente molto diversi e, in pratica, sono due modelli ben distinti, mentre A500 e A500+ non hanno differenze notevoli tra loro. L'A2000A si distingue immediatamente dall'A2000B poiché è privo del connettore video monocromatico RCA.

Filosofie costruttive

I personal si possono empiricamente classificare in due architetture sulla base del modo in cui vengono sincronizzati i componenti del sistema: clock legato al video o clock legato alla CPU. Alla seconda categoria appartengono, per esempio, gli IBM compatibili. In essi le funzioni di I/O e anche video sono svolte da schede dedicate poste su un bus di espansione (più lento di quello che collega CPU e memorie). Questo sistema è semplice da progettare e da espandere, ma ha il grande svantaggio che le prestazioni grafiche sono molto penalizzate. La memoria video è dedicata a questo solo scopo, è disponibile in banchi posti a locazioni fisse ed, in pratica, è solo un "framebuffer" in cui il processore scarica la bitmap di un'immagine avendo un'idea molto vaga di che cosa viene visualizzato istante per istante. L'accesso a questa RAM video, inoltre, è molto lento, poiché (a meno che non si usino delle memorie di tipo VRAM, molto costose) ci sono grossi problemi di sincronizzazione tra CPU e controller della scheda video.

Se la scheda video è invece di tipo "intelligente" (per esempio, TIGA) la velocità può essere notevole, ma il processore non ha più la minima idea di che cosa stia visualizzando la scheda ad un dato momento. In pratica, questa architettura è adatta alla gestione di immagini statiche o di testo. Alla prima categoria appartengono i vecchi home computer a 8 bit. La funzionalità video è integrata sulla motherboard e il clock del microprocessore, e di conseguenza anche quello del bus di espansione, è ricavato da quello dei circuiti video.

Ciò comporta molti vantaggi: innanzitutto CPU e circuiteria video sono sempre sincronizzati tra loro, perciò possono spartirsi in buona amicizia il bus per l'accesso alla RAM video (vedremo meglio in seguito come ciò avviene su Amiga). Inoltre, la CPU può sapere in ogni momento che cosa sta facendo la circuiteria video; questo approccio non limita poi in nessun modo le capacità di espansione sul bus, sul quale possono risiedere altre schede video che genereranno segnali, automaticamente sincronizzati con quelli del video sulla motherboard. Questa capacità è essenziale nell'ambito videografico.

Lo scotto da pagare è la necessità di far girare la CPU ad una "strana" frequenza di clock che in genere è inferiore a quella massima ammessa dal costruttore per quel chip; gli aumenti di velocità possono avvenire solo a passi discreti (per esempio, 7, 14, 28 MHz). Inoltre, la stretta integrazione di tutte le principali funzioni nella motherboard rende più

difficile la riparazione e più facile un guasto catastrofico.

Anche gli Amiga inespansi sono fatti così, ma con una sostanziale e geniale differenza: è possibile separare il bus usato dai circuiti video da quello usato dalla CPU, usando il "CPU slot" come porta di collegamento. Questo significa che, con una scheda acceleratrice, l'architettura del computer si modifica in modo da avere due bus distinti: uno ottimizzato per i circuiti video e uno ottimizzato per le esigenze della CPU, con un incremento di prestazioni superiore a quello che si otterrebbe se ci si limitasse a sostituire il processore con uno più potente.

In questo modo ciò che sta sulla motherboard è sincrono con la circuiteria video, e ciò che sta sulla scheda acceleratrice, invece, può girare al massimo della velocità consentita per quella CPU. E' anche una soluzione geniale per quanto riguarda i costi di produzione: il sistema base contiene solo i componenti economici a bassa velocità mentre tutte le parti pregiate ad alta velocità risiedono sulla scheda acceleratrice che è disponibile come opzione.

Nell'A3000 e A3000T questo concetto è ulteriormente sviluppato, ponendo una prima "porta di collegamento" tra i bus dentro un nuovo chip custom, e lasciando in questo modo girare a 25 (o 16) MHz anche gli slot e il controller hard disk, mentre lo slot CPU funge da "seconda porta di collegamento" a cui è possibile collegare dispositivi ancor più veloci come coprocessori vettoriali o processori a 50 MHz. E' da notare che una organizzazione di questo tipo non esiste neppure nella maggior parte delle macchine di classe workstation, oltre che su nessun altro personal.

Un'altra peculiarità di A2000B, A3000, A3000T è lo slot video. Esso non è (come a prima vista si potrebbe pensare) un connettore per schede video (nel senso proprio del termine), ma è un connettore su cui sono riportati i principali segnali della circuiteria video e della porta parallela, che dà la possibilità di manipolare ulteriormente il segnale video generato da Denise in modo da aggiungere nuove prestazioni come genlock o più colori. E' davvero un concetto semplice e di costo realizzativo quasi nullo, ma pressocché nessun altro personal lo ha! Sull'A2000A si tratta invece di un semplice riporto dei segnali del connettore per il monitor.

I chip custom

I computer della famiglia Amiga sono basati, come è noto, su un insieme di chip custom di tipo LSI (Large Scale Integration). Ciò significa che per gli standard moderni sono dei chip piuttosto piccoli, con meno di 100.000 transistor MOS.

Dal punto di vista costruttivo, per quello che è possibile sapere dalle informazioni pubblicate verso il 1985 (pertanto oggi potrebbero essere superate), Denise, Paula, Gary e Fat Agnus sono realizzati in tecnologia N-HMOS con canale di 3 micron, cioè una tecnologia veramente vecchia che rende molto difficile effettuare cambiamenti, soprattutto per Denise che ha un'area vicina ai 64mm quadrati che sono il massimo per questa tecnologia.

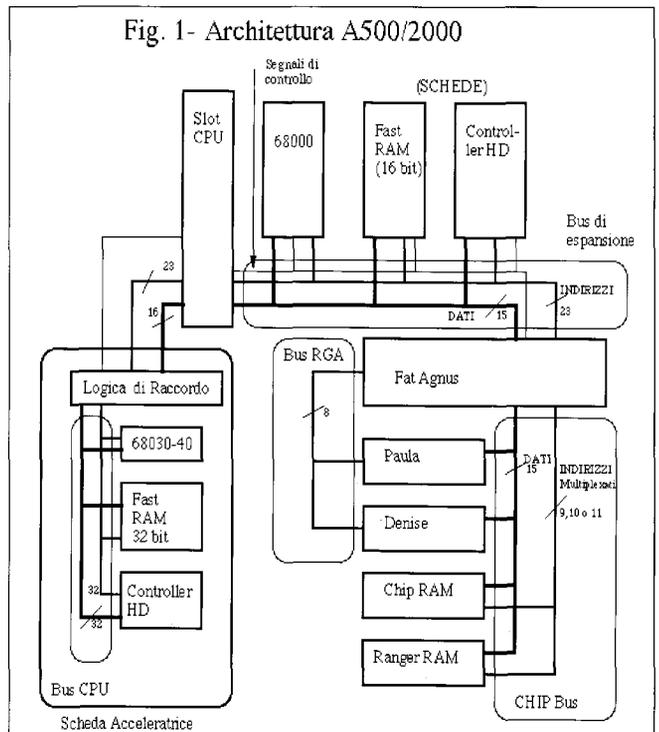
Futuri miglioramenti di questi chip dovrebbero quindi necessariamente passare per un completo rifacimento in tecnologia più moderna, con il vantaggio che la complessità potrebbe essere notevolmente superiore a quella attuale e che l'espandibilità, soprattutto, incontri meno problemi.

Questo è lo scotto da pagare per avere dei chip full custom: come in un programma scritto in assembler ottimizzato, è possibile spremere il massimo dalle risorse presenti, ma effettuare piccoli cambiamenti è problematico. E' più conveniente rifare tutto quando le nuove tecnologie disponibili consentono miglioramenti: si ha un tipico avanzare "a grossi balzi". Il tempo di sviluppo di un chip full custom si aggira sui 2-4 anni, perciò un progetto (come Amiga) che ne fa uso può essere rinnovato solo dopo questo periodo di tempo.

Invece gli IBM compatibili, che fanno uso di una tecnologia basata su gate array, consentono un graduale aumento di prestazioni nel tempo, ma non sarà mai possibile sfruttare al 100% il silicio di quei chip. Tutti i nuovi chip di A3000 e A3000T sono gate array.

I bus

Su Amiga sono presenti diversi bus, interconnessi tra loro da elementi di collegamento (Figura 1).

Fig. 1- Architettura A500/2000

Ricordo brevemente che un bus è un insieme di fili condiviso da un certo numero di dispositivi, perciò occorre sempre un "arbitro" che stabilisce quale dispositivo è autorizzato ad accedere al bus in un certo istante.

I nomi che userò sono solo di comodo, in quanto non esiste ancora un accordo nella letteratura tecnica su Amiga per quanto riguarda la nomenclatura. Altrove potrete, per esempio, sentir citare termini come "local bus" e "system bus", per analogia con i sistemi Intel e i data book Motorola.

Un primo bus è il "chip bus". E' accessibile sul connettore interno (lo sportellino) dell'A500-A500+. Questo bus è arbitrato da Fat Agnus ed è ottimizzato per le necessità del video e degli altri chip custom; gli indirizzi sono presenti in forma multiplexata per poter essere direttamente connessi a uno o due banchi di RAM dinamiche.

I chip custom hanno bisogno di poter accedere alla memoria a intervalli rigidamente fissati, perciò questo bus è anche detto "di tipo deterministico", in quanto ciascun dispositivo è certo di poter leggere o scrivere i suoi dati a cadenza fissa. Quando la CPU accede a questo bus deve attendere pazientemente che si "liberi un posto", come un pedone che deve attraversare una strada con molto traffico.

Poiché questo bus deve essere sincrono con il clock dei circuiti video, eventuali aumenti della sua velocità dovranno necessariamente restare multipli della attuale cadenza di prelievo dei dati dalla RAM (cioè in pratica della frequenza di clock, 7.15 MHz per le macchine NTSC e 7.09 MHz per le macchine PAL) e della larghezza di parola (16 bit): lo vedremo meglio quando parleremo dell'Agnus. Di conseguenza il numero di bitplane o la loro dimensione potrebbero solo raddoppiare, quadruplicare, ottuplicare...
Il chip bus è connesso al bus di espansione tramite una circuiteria contenuta in Fat Agnus, che collega anche l'RGA bus agli indirizzi del bus di espansione.

Un secondo bus è l'"RGA bus": si tratta di un bus usato dai chip custom per comunicare tra loro e con la CPU (nel modo che vedremo). E' simile ad un bus di indirizzi, è a 9 bit (il bit 0 al solito non è usato) e non è accessibile all'esterno degli attuali Amiga. Non è un vero bus di indirizzi poiché ciascun chip custom conosce direttamente a quali indirizzi rispondere senza bisogno di circuiti di decodifica. Intervenendo su questo bus sarebbe possibile alterare anche di parecchio le prestazioni del computer.

Un terzo bus è il "CPU bus", che è ottimizzato per le necessità della CPU. Esso dunque cambia al cambiare del microprocessore utilizzato, supportando, di volta in volta, il numero di stati di attesa, la larghezza di parola e il tipo di accesso alla RAM (burst, page mode...) più adatto. Nell'A500/A2000 è interno alle schede acceleratrici, e su di esso risiedono la fast RAM a 32 bit e l'eventuale controller per hard disk "veloce". Si interfaccia fisicamente al bus di espansione per mezzo dello slot CPU dell'A2000B, oppure in modo assai più "rozzo" per mezzo del connettore laterale dell'A1000, dell'A500 o dello slot CPU dell'A2000A (molto più semplificato di quello dell'A2000B).

Il bus di espansione è il bus disponibile sugli slot degli A2000, e sul connettore laterale dell'A500 e A1000. Nell'A500 e A1000 si chiama "bus Zorro", dal nome del primo prototipo di Amiga che lo possedeva, contenuto in una scatola metallica nera che oggi è nell'ufficio di Dave Hayne (il papà del Commodore 128 e di Amiga 3000) a West Chester.

Anche se la Commodore non supporta l'uso delle espansioni da collegare al connettore laterale dell'A500 sull'A1000 e viceversa, in realtà i due connettori sono elettricamente e dinamicamente perfettamente identici (con l'eccezione di un segnale, che su A500 è non connesso).

Nell'Amiga 2000 è chiamato Zorro II, nell'Amiga 3000 è lo Zorro III (compatibile con lo Zorro II). Zorro II e Zorro sono a 16 bit e i soliti 7 MHz circa (che consentono una velocità di 3.5 Mb/s, paragonabile ai 4 Mb/s del bus ISA); Zorro III è asincrono e a 32 bit, consentendo una velocità di trasferimento dei dati di addirittura 20 Mb/s random e 33 Mb/s burst, con massimi teorici rispettivamente di 50 Mb/s e 150 Mb/s.

I bus Zorro I e Zorro II sono molto simili al bus del 68000 stesso, mentre lo Zorro III non è modellato sul bus del 68030 bensì, per garantire la massima compatibilità con le architetture più moderne, è uno standard a sé adattabile ad ogni processore (anche non Motorola).

Nell'A2000B questo bus è arbitrato dal gate array (equivalente a circa 150 porte logiche) Buster, nell'A3000 e A3000T dal Fat Buster, negli altri modelli ci si affida alla logica di arbitraggio interna del 68000. In questo bus si installano le schede di I/O e le espansioni di fast RAM a 16 bit. Si tratta di un bus sincrono con il clock video, ma ottimizzato per le esigenze della CPU invece che per i circuiti video. Zorro e Zorro II non sono deterministici: un dispositivo che faccia una richiesta di accesso al bus dovrà attendere finché non è libero. Infine, ci sono alcuni bus di input/output seriali, come quello della tastiera o del disk-drive, per noi di scarso interesse.

La separazione del chip bus dal bus di espansione consente un aumento di prestazioni, poiché mentre la CPU (o un dispositivo DMA su scheda) accede al bus di espansione o al CPU bus, Fat Agnus può accedere contemporaneamente ad un'altra cella di memoria posta sul chip bus. Questo comporta, però, anche la necessità di una forma di sincronizzazione tra bus di espansione e chip bus, affinché la CPU possa accedere alla memoria sul chip bus senza entrare in conflitto con una operazione già in corso da parte di Fat Agnus.

Occorre anche sincronizzare tra loro CPU bus e bus di

espansione, e ciò è effettuato essenzialmente dalla logica interna alla scheda acceleratrice.

Uno sguardo al 68000

Vediamo rapidamente quali sono le temporizzazioni essenziali del microprocessore MC68000, che ci serviranno per capire meglio come funzionano i bus.

Il ciclo di lettura (figura 2) si svolge in questo modo: nella fase S1 vengono emessi gli indirizzi, nella fase S2 è attivata l'abilitazione degli indirizzi /AS; ora il microprocessore attende sino all'inizio dello stato S5 per leggere il valore della linea /DTACK: se in quel momento è a zero logico, procede alla lettura dei dati, altrimenti attende che sia a livello logico zero, inserendo degli stati di attesa ("wait-state") sempre in numero pari.

Se non sono stati inseriti dei wait-state, alla fine della fase S6 viene riportata alta l'abilitazione degli indirizzi /AS e il dispositivo che aveva generato /DTACK deve riportarlo immediatamente a livello alto, mentre le periferiche sul bus devono riportare le linee dei dati in alta impedenza.

Possiamo già notare alcune particolarità del 68000 che vengono sfruttate da Amiga. Innanzitutto, negli Amiga 500, 1000 e 2000 normalmente il segnale /DTACK viene asserito da Fat Agnus in modo da ottenere sempre 0 wait state ad OGNI ciclo di bus della CPU. Ciò comporta che tale segnale venga generato anche se la locazione a cui il programma ha fatto accesso non esiste.

Il processore non è in grado di accorgersi di questo fatto anomalo, e prosegue come se niente fosse. Un programma che abbia questo bug, su un Amiga con 68000 può funzionare apparentemente a meraviglia, su un Amiga accelerato potrà comportarsi in modo casuale, che dipende da come è realizzata l'acceleratrice stessa. In genere, con un 68020/68030 è possibile ricreare le condizioni presenti su un Amiga standard e impedire "deragliamenti", ma questo è quasi impossibile sia sull'A3000 (dove il segnale /DTACK deve essere generato dalla singola scheda di memoria), sia con il 68040 (a causa della sua pipeline).

Fig. 2 - Ciclo di lettura di word MC60000 (senza wait-state)

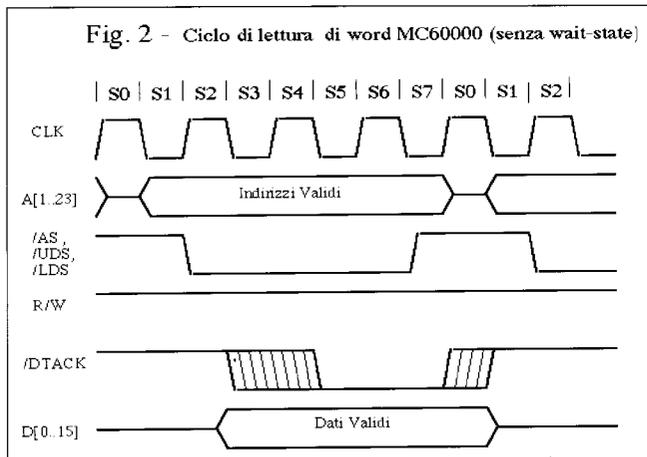
Questa pecca dell'A1000, poi propagatasi per compatibilità su A500 e A2000, è la causa di molte delle incompatibilità dei programmi con le schede acceleratrici. Su A3000 è presente un apposito circuito per recuperare questo tipo di bug, ma funziona solo quando non si entra in conflitto con le esigenze del bus Zorro III. A parte questo, la sincronizzazione tra bus di espansione e chip bus diventa un problema banale: semplicemente, quando la CPU emette un indirizzo mappato nel chip bus, Fat Agnus (che è l'arbitro del chip bus) aspetta a mandarle il segnale /DTACK finché non ha finito la sua operazione e, al tempo stesso, mantiene aperti i buffer di collegamento tra chip bus e gli altri bus.

Notiamo anche un'altra particolarità di questa architettura, Fat Agnus genera il segnale /DTACK anche per i dispositivi che non sono posti sul chip bus da lui controllato: ciò fa sì che di default ogni memoria o I/O deve funzionare a 0 wait, su qualsiasi bus sia. Se sono richiesti dei wait-state, il dispositivo deve attivare il segnale XRDY entro 60 nanosecondi dalla transizione alto/basso di /AS, oppure deve disabilitare il generatore di /DTACK dell'Agnus tramite il segnale /OVR, e creare da sé il segnale /DTACK.

Per lo stesso motivo, su A500 e A2000 non è in genere possibile eseguire l'istruzione TAS: essa genera un ciclo di bus completamente diverso, sia da quello di lettura che da quello di scrittura, e Fat Agnus tenta di interromperlo nel mezzo, causando un errore di bus e quindi il blocco del sistema. Alcune schede acceleratrici sono consapevoli del problema e trasformano il ciclo di bus dell'istruzione TAS in due cicli normali; però solo su A3000 il problema è stato risolto alla radice.

Il segnale /DTACK e gli indirizzi sono riportati sul bus di espansione attraverso dei buffer, che hanno un ritardo di propagazione. Di conseguenza le schede hanno pochissimo tempo per rendere disponibili i dati durante le operazioni di lettura; spesso viene usato il segnale /OVR per poter generare localmente /DTACK e averlo così disponibile in tempo.

Senza scendere in ulteriori dettagli, si può notare come la temporizzazione sia delicatissima, soprattutto per il DMA, sul bus di espansione e pertanto le schede devono essere progettate con estrema attenzione per entrare nelle specifiche e non provocare malfunzionamenti di altre schede poste sul bus di espansione. Se, per esempio, una scheda generasse un segnale /DTACK leggermente in ritardo, altre schede, che contano sul fatto che esso arrivi entro un certo periodo di tempo, cessano di funzionare.

Ecco perché le schede particolarmente economiche, specialmente se sono usate su A1000, A500 e A2000A, che sono più delicati come capacità di pilotaggio, sono da guardare con sospetto in quanto se sono state progettate "prendendo

delle scorciatoie" possono risultare incompatibili con altre espansioni presenti.

Il sintomo caratteristico di questo genere di problema è il computer che non parte proprio (schermo che resta nero), oppure che funziona per un po' e poi si "congela", senza guru, finché non si provvede al reset. Se vengono usate delle porte TTL in una scheda, esse devono necessariamente essere veloci, possibilmente della serie F (fast). Tranne che per alcuni segnali, è assolutamente esclusa la possibilità di usare delle comuni porte serie LS o HCT come si fa sugli IBM compatibili: se volete autocostruirvi delle schede, fate attenzione!

Ecco anche perché l'idea originaria di espansione dell'A1000, che prevedeva una catena di dispositivi dotati di connettore pass-trough collegati in cascata al connettore laterale, è stata quasi subito abbandonata: il ritardo di propagazione introdotto dai buffer sulle linee dati, indirizzi, /AS e /DTACK di ciascun dispositivo rende pressocché impossibile operare con 0 wait state già per il secondo dispositivo. Bisogna comunque doverosamente aggiungere che lo slot laterale dell'A1000 è stato progettato male, e i segnali sono inquinati da rumore. Fortunatamente, la Commodore ha introdotto a partire dalla revisione 1.2 del sistema operativo il supporto ad una nuova tecnica di espansione.

Amiga Expansion Architecture

Uno dei punti di forza della famiglia Amiga è la sua architettura di espansione. Elettricamente, i bus Zorro sono composti da PIC (Plug In Card, cioè schede) e Backplane. I backplane sono invisibili al software e contengono tutti i circuiti per pilotare correttamente le schede e interfacciarsi al microprocessore. In tutte le correnti implementazioni, si usa un solo backplane che risiede sulla motherboard. Esistono però per A500 degli scatolotti che contengono un backplane sul quale si possono collegare un certo numero di PIC: non si tratta dell'invenzione di qualche intraprendente produttore, ma la possibilità è esplicitamente prevista.

Tutte le PIC devono supportare il protocollo di autoconfigurazione "Autoconfig". E' un metodo per evitare all'utente di dover effettuare complessi settaggi ogni volta che si cambia la configurazione hardware della macchina, inserendo o togliendo risorse sul bus di espansione. E' una soluzione senza dubbio più elegante e funzionale sia del metodo "a jumper", usato sui primi personal e anche sui computer IBM compatibili XT, sia del metodo a "RAM CMOS", usato, per esempio, sui personal IBM di classe AT, in cui la configurazione è impostata manualmente dall'utente per mezzo di un programma (ed è facilmente rovinata da virus o pasticcioni alla tastiera...).

Non voglio in questo ambito discutere approfonditamente il protocollo, in quanto una completa descrizione si può trovare nei due libri che ho già citato ed è una complessa interazione tra hardware e software che riempirebbe 50 pagine. Non è però magia, perciò i produttori di schede che non lo implementano o, peggio, dichiarano di averlo implementato quando in realtà non è vero, non hanno nessuna scusante.

Quello che è importante notare è che per lo Zorro II sono previsti al più 8 Mb di spazio di indirizzamento, più alcune decine di kilobyte per dispositivi di I/O, essendo 24 i bit di indirizzo disponibili sul connettore del bus di espansione (o meglio: il bit zero non è riportato, in accordo all'architettura del 68000) e 8 Mb sono riservati ad altro scopo. La RAM che risiede sul chip bus, cioè chip RAM e "ranger" RAM, non entra a far parte del conto.

Nelle macchine attuali, sarebbero disponibili al più 2 Mb di chip RAM e 1.75 Mb di ranger RAM, per cui il massimo teorico di RAM installabile è 11.75 Mb. Su A3000 e A3000T invece lo spazio riservato alle espansioni Autoconfig è di 1750 Mb e quello allo slot CPU è di 128 Mb. AmigaOS può usare al più 31 bit di indirizzo e quello è il limite filosoficamente equivalente ai 640 K di un altro celebre sistema operativo.

La Commodore, comunque, ufficialmente supporta solo 0.5 Mb di ranger RAM (detta anche slow fast); essa è RAM mappata a partire da C00000 esadecimale e pur non essendo accessibile per il DMA di Fat Agnus come la fast RAM, in quanto è posta sul chip bus, è soggetta agli stessi problemi di accesso della chip RAM.

Però la fast RAM a 32 bit, che risiede fisicamente sul CPU bus, secondo le specifiche Autoconfig, deve entrare a far parte degli 8 Mb complessivi, in quanto la logica di interfaccia tra CPU bus e bus di espansione deve consentire a qualsiasi scheda (tipicamente un controller per hard disk, ma potrebbe essere una scheda video a 24 bit) posta sul bus di espansione l'accesso alla RAM sul bus CPU.

Per superare la barriera degli 8 Mb di fast RAM, oppure proprio per risparmiare le poche lire aggiuntive che una implementazione completa richiede, alcuni produttori di schede acceleratrici consentono di installare della RAM a 32 bit non autoconfigurante, che perciò non ha in pratica limiti come quantità. Questo è in disaccordo con le indicazioni della Commodore.

Abbiamo visto qual è l'architettura generale dei nostri Amiga, la prossima volta vedremo qualcosa di un po' (ma solo poco) più complesso: come funzionano i chip custom.

(segue a pag. 51)

Capitolo 4

AUDIO

Le capacità sonore di Amiga sono notevolmente superiori alla media degli altri personal computer; la motivazione principale risiede nel fatto che Amiga genera il suono in modo digitale, come fa Macintosh, e non analogico, come il resto dei personal computer. Per comprendere come Amiga produce un suono conviene fare un passo indietro.

Suono analogico e digitale

I suoni che percepiamo sono prodotti dalla variazione della pressione dell'aria. L'onda sonora che rappresenta questa variazione continua è riportata in figura 4.1. Una tipica forma d'onda sonora è sinosuidale ed è quella che verrà utilizzata per gli esempi di questo capitolo. Nel grafico di figura 4.1 l'asse delle ascisse indica il tempo e quello delle ordinate la variazione dell'ampiezza della forma d'onda.
Il problema che si pone a questo punto è questo: come può Amiga rappresentare una variazione continua di tensione (poiché i suoni vengono trasformati in tensione dai microfoni), quindi una grandezza analogica, se lavora in modo digitale, ossia solo con 0 e 1?
La risposta a questo quesito è abbastanza semplice. Dalla forma analogica viene letto il valore di tensione a intervalli di tempo molto piccoli; alla fine si otterrà una tabella con una serie di numeri (grandezza digitale).

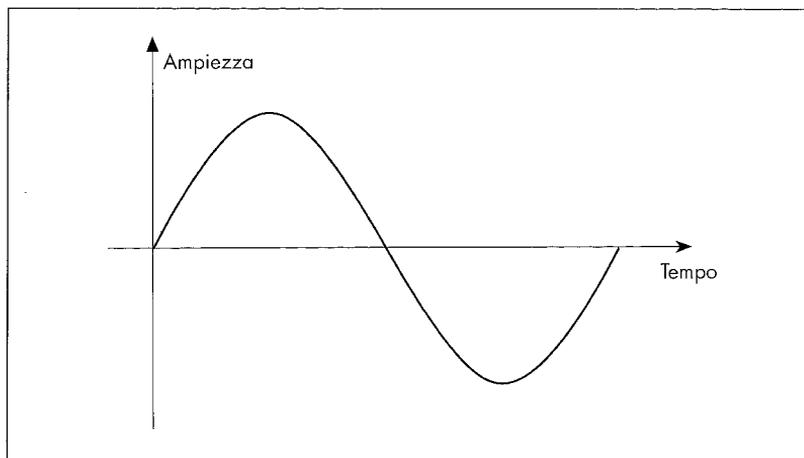



Figura 4 1
Una forma d'onda sonora

Questo processo di ***digitalizzazione*** (ossia di conversione da una grandezza analogica a una digitale) viene chiamato ***campionamento*** e viene effettuato da appositi circuiti chiamati ***ADC*** (Analog/Digital Converter) o convertitori A/D.

Se poi si inviano i valori appena ottenuti con gli stessi intervalli di tempo a un convertitore D/A, ***DAC*** (Digital/Analog Converter) si ottiene una forma d'onda simile a quella rappresentata in figura 4.2.

Lo scopo del processo di campionamento è, quindi, quello di ottenere un valore costituito da un certo numero di bit che rappresenti il valore di tensione in un istante specifico della forma d'onda analogica. È chiaro che più è elevato il numero di bit che rappresenta il valore di tensione più ampia sarà la gamma dinamica (differenza tra il valore più alto e quello più basso) campionabile. Un altro parametro da tenere in considerazione in questa fase è la ***frequenza di campionamento***, ossia l'intervallo di tempo tra un campionamento e l'altro: più è elevata la frequenza (ossia più breve è l'intervallo) e maggiore sarà la fedeltà del suono campionato.

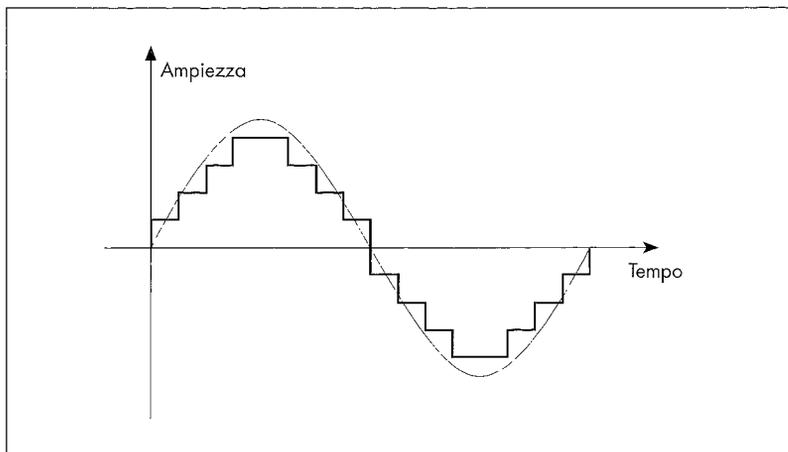



Figura 4.2
Forma d'onda ottenuta dopo il campionamento

Il compact disc, che funziona in modo digitale (il laser che legge il disco restituisce 0 o 1), possiede dei convertitori D/A a 16 bit, permettendo così di ottenere 65536 valori diversi, e utilizza una frequenza di campionamento di poco superiore ai 44 KHz (in un secondo vi sono più di 44000 letture dal disco).

L'audio di Amiga

Amiga, nel suo piccolo, possiede quattro convertitori D/A a 8 bit (256 valori diversi) e ha una frequenza massima di campionamento pari a 28,867 KHz. Questi quattro convertitori sono programmabili indipendentemente l'uno dall'altro e possiedono, ciascuno, un canale audio di uscita ben definito. I canali audio, numerati da 0 a 3, sono accoppiati per ottenere un'uscita stereo: i canali 1 e 2 formano l'uscita di destra, mentre i canali 0 e 3 l'uscita di sinistra. Come è stato accennato nei primi capitoli, i quattro convertitori si trovano fisicamente all'interno di Paula e a ognuno di essi è associato un canale DMA.

Esistono due metodi per la creazione di un suono: il primo, e il più utilizzato, è detto *automatico* e il secondo è detto *manuale*.

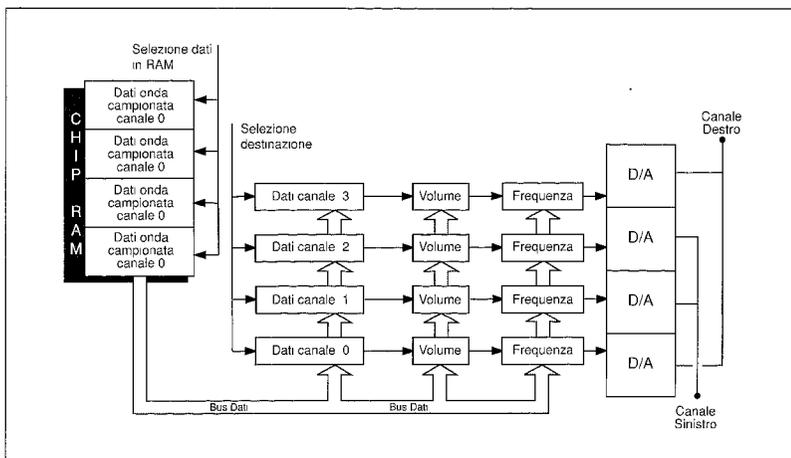



Figura 4.4
Circuiteria interna dei canali DMA

Il sistema sonoro può quindi essere visto come un puntatore in memoria (gestito dal sottosistema sonoro) che preleva i valori a una velocità ben definita (impostata sempre dal sottosistema sonoro) e li spedisce al convertitore D/A per convertire in tensione i valori numerici, affinché possano essere inviati agli altoparlanti. Naturalmente, visto che i quattro canali audio sono programmabili in modo indipendente l'uno dall'altro, è possibile generare quattro diversi segnali audio contemporaneamente.

È importante specificare che il metodo automatico di creazione di un suono prevede la descrizione di un ciclo completo della forma d'onda. Fatto questo passo il suono viene generato senza altri interventi esterni da parte dell'utente.

Nella figura 4.3 viene riportato lo schema di funzionamento per la creazione di un suono, mentre nella figura 4.4 è rappresentata schematicamente la circuiteria interna relativa ai canali audio.

Metodo manuale

A differenza del metodo automatico, quello manuale prevede la descri-

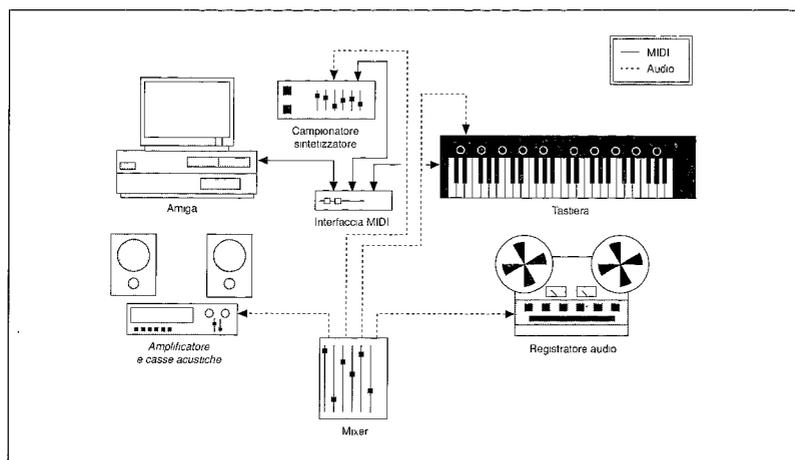



Figura 4.5
Sistema Amiga-MIDI

zione istante per istante dell'onda sonora da generare. L'utente deve quindi scrivere direttamente nei convertitori D/A i valori calcolati in base alla tabella in memoria. In questo modo si possono ottenere dei risultati sonori impensabili; il grande difetto però è che questo metodo sottrae molto "CPU time", ossia impegna molto la CPU (in quanto non vengono utilizzati i canali DMA ma la CPU per scrivere i dati nei convertitori) per cui viene scarsamente utilizzato.

Il MIDI

L'ultima sofisticazione tecnologica nel campo musicale è il MIDI (Musical Instrument Digital Interface), un linguaggio e un insieme di protocolli che permettono a dispositivi musicali MIDI compatibili di comunicare tra di loro. Amiga può facilmente diventare il centro di un sistema musicale MIDI collegando un'interfaccia MIDI alla porta seriale e i dispositivi MIDI all'interfaccia, come mostrato in figura 4.5. Una volta che Amiga è stato "MIDIzzato", è possibile suonare su di uno strumento MIDI e vedere le note apparire sullo schermo di Amiga, utilizzando, ovviamente, il software adatto.

Un'altra possibilità è quella di creare dei suoni con Amiga e utilizzarli con una tastiera MIDI compatibile. Qualsiasi interfaccia MIDI offre molti canali MIDI e porte MIDI in/out per il collegamento di apparecchiature MIDI.
Dopo Amiga e l'interfaccia MIDI, il pezzo più importante in un sistema MIDI è una tastiera MIDI compatibile. Alcuni programmi permettono di utilizzare la tastiera di Amiga come una tastiera musicale, anche se ciò non è raccomandabile neanche per i principianti. Sul mercato si trovano tastiere MIDI compatibili anche a basso prezzo.

Le tastiere possono anche contenere il proprio campionatore e sintetizzatore. Recentemente sono stati resi "MIDI compatibili" anche equipaggiamenti tipo ottoni, percussioni, strumenti a corda e a fiato; ovviamente è possibile utilizzare uno qualsiasi degli strumenti MIDI, non solo le tastiere.

Software MIDI

Come è appena stato detto, per poter utilizzare un'interfaccia MIDI con i relativi strumenti è necessario utilizzare del software specifico. Questi si dividono per tipi a seconda delle funzioni che svolgono:

- *sequencer*, per il controllo del suono proveniente dai dispositivi MIDI;
- *sintetizzatori*, chiamati anche synth, per la creazione di suoni e musica;
- *drum synthesizer*, per effetti di percussione automatici;
- *note editor*, (notazione musicale) per comporre, modificare, trascrivere e suonare le partiture musicali;
- *patch editor/librarian*, per il collegamento a equipaggiamenti audio (un patch è la connessione tra due o più tracce o sorgenti audio). Un patch editor è specifico per lo strumento musicale collegato con esso.

La maggior parte del software musicale di Amiga è MIDI compatibile, come Sonix, per esempio. Se si ha intenzione di utilizzare professionalmente il MIDI come strumento di lavoro, allora è necessario disporre di

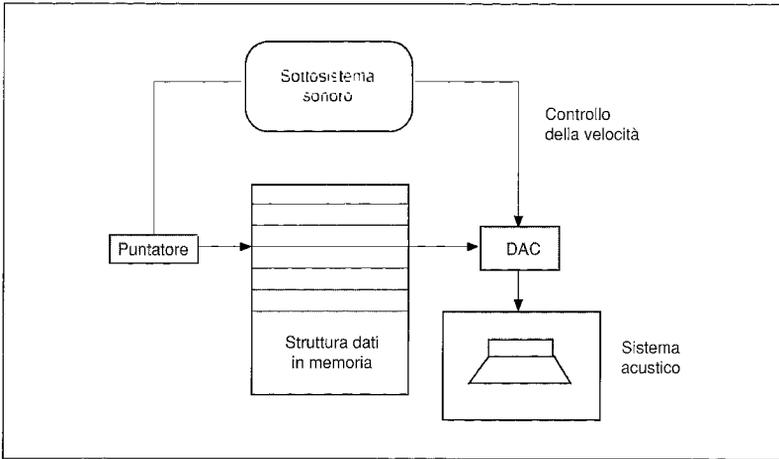



Figura 4.3
Schema di funzionamento per la creazione di un suono

Creazione di un suono

Metodo automatico

La creazione di un suono con il metodo automatico prevede alcuni passi ben definiti:

- decidere quale canale DMA utilizzare (si ricordi che questa scelta deve essere fatta in base al canale stereo sul quale di intende udire il suono);
- definire la forma d'onda sonora e creare conseguentemente in memoria un'apposita struttura dati;
- informare il sistema dove poter reperire questi dati;
- impostare alcuni registri specifici che comunichino a un sottosistema sonoro dove reperire i dati (il sottosistema sonoro ha il compito di gestire alcuni parametri sonori, come la velocità di lettura dei dati, che modificano notevolmente il suono finale);
- scegliere il volume;
- impostare la frequenza di campionamento;
- selezionare il canale audio e far partire il DMA.

un sequencer MIDI. Tra questi prodotti si segnalano Bar & Pipes1.0 e la relativa versione *Professional*, KCS 3.5 e KCS Level II 3.5, Master Track Pro, MIDI Recording Studio, Music X, Pro 24 III, Pro MIDI Studio, Studio 24, Tiger Cub, Track 24 e Trax.

Per quanto riguarda i programmi di notazione musicale sono disponibili The Copyist III, Deluxe Music Construction Set 2.5 e Studio 24. Tra gli editor/librarian per sintetizzatori si segnalano, Super Librarian, Casio CZ Rider, Casio VZ Rider, Emu Proteus 1/XR, Esqapade, Lexicon PCM70, Kawai K1, Kawai K5, K1 Master Editor/librarian, Korg M1, Oberheim M-6 Tricks, Oberheim Matrix 6/1000 Master editor/librarian, Roland D-10/110/20, Roland D-10/110/20 Master editor/librarian, Roland D-50/550, Roland MT-32, Yamaha DX Heaven, Yamaha DX7II/TX802, 4-Op Deluxe. Il mercato degli editor/librarian per campionatori offre, invece, MIDI Sample Wrench, Synthia II e Synthia Professional.

Per tutti i riferimenti a questi prodotti si consiglia di consultare le riviste del settore di informatica musicale.

L'ultima cosa da precisare è che, come tutti gli altri settori, anche quello dell'informatica musicale è in continuo sviluppo; probabilmente quando saranno lette queste righe vi saranno una decina di prodotti nuovi o, comunque, aggiornati.

Il serial device

Un esempio completo

Antonello Biancalana

Antonello Biancalana lavora per la ProMIND, una software house di Perugia che sviluppa software grafico e musicale per Amiga, ed è sviluppatore Amiga registrato nella categoria "commercial". Di recente, Antonello Biancalana, ha progettato e sviluppato MSPL (Music Synthesis Programming Language), un particolare linguaggio di programmazione rivolto alla sintesi sonora e musicale.

Amiga possiede una porta seriale compatibile RS-232C, che gli permette di dialogare con il mondo esterno e che può essere usata facilmente in tutte le applicazioni che richiedono l'invio e la ricezione di flussi di dati (modem, MIDI, collegamenti fra computer, ecc.).

L'accesso alla porta seriale può avvenire in due modi diversi, a secondo dell'uso che si intende fare del dispositivo. Infatti, la porta seriale può essere gestita attraverso i registri hardware, oppure nella modalità standard prevista dall'ambiente Amiga, cioè facendo uso dei "device".

Il primo metodo viene usato raramente, solo quando si ha bisogno del dispositivo seriale per lavorare ad alte velocità: ovviamente, questo tipo di implementazione deve avvenire in linguaggio assembler. Il secondo metodo è quello che viene usato nelle applicazioni più "comuni", in quanto la gestione dei device fornita da Exec semplifica notevolmente l'uso della porta seriale.

Questo articolo si occupa esclusivamente del secondo metodo, facendo quindi riferimento al cosiddetto "serial.device". Gli esempi, per motivi di chiarezza, saranno in linguaggio "C"; comunque, chi volesse accedere al "serial.device" in assembler, non dovrebbe incontrare difficoltà nel gestire correttamente il dispositivo, utilizzando le informazioni riportate.

L'inizializzazione

Come abbiamo già affermato in precedenza, il "serial.device" di Amiga viene gestito mediante Exec, pertanto, dovremo fare riferimento a tutte le convenzioni previste per la gestione dell'Input/Output dei device.

Come per tutti i dispositivi di Amiga, la prima cosa da fare è quella di provvedere alla sua apertura. Il "serial.device" richiede l'uso di una struttura dati di tipo "IOExtSer" e non della struttura "IOStdReq", tipica dei dispositivi standard di Exec. Avremo quindi bisogno di una struttura di questo tipo, propriamente inizializzata, prima di poter aprire il "serial.device".

Per farlo, utilizzeremo la funzione CreateExtIO( ): è definita all'interno di "Amiga.lib", che deve dunque essere aggiunta al codice oggetto generato dal compilatore durante la fase di linking. La funzione CreateExtIO( ) necessita di due parametri: l'indirizzo della porta da usare nella gestione del dispositivo e la dimensione della struttura dati che intendiamo creare. Possiamo ottenere la porta di cui abbiamo bisogno, mediante la funzione CreatePort( ) di Amiga.lib:

```
struct MsgPort *MyPort;
MyPort = CreatePort("MySerialPort",0);
```

La funzione "CreatePort" necessita di due parametri: il primo rappresenta il nome da associare alla porta, mentre il secondo indica la priorità. Il nome della porta non è obbligatorio, ma è necessario nel caso in cui vogliamo che altri task siano in grado di trovarne l'indirizzo mediante la funzione FindPort( ).

Nell'esempio precedente manca un elemento importantissimo, che dovrà essere presente nel codice finale: il controllo dell'effettiva creazione della porta. E' necessario controllare se Exec sia riuscito a creare la porta. Questo è decisivo perché, quando successivamente faremo riferimento ad una porta inesistente (o a qualsiasi altra cosa), Amiga ci saluterà con una magnifica "Guru meditation".

Una volta ottenuta la porta, dobbiamo creare la struttura "IOExtSer" mediante la funzione CreateExtIO( ):

```
struct IOExtSer *SerialData;
SerialData = (struct IOExtSer *)CreateExtIO(
        MyPort,sizeof(struct IOExtSer));
```

Questa funzione allocherà la memoria e i dati necessari alla nostra struttura: da questo momento potremo cominciare ad occuparci del dispositivo seriale.
La funzione CreateExtIO( ) ritorna un puntatore a una struttura di tipo "IORequest", che è ovviamente diversa

dalla nostra struttura "IOExtSer".

L'operatore di "casting" (struct IOExtSer *), posto prima della funzione CreateExtIO( ), viene utilizzato per convertire correttamente il puntatore ritornato dalla funzione con quello specificato, ovvero con un puntatore alla struttura "IOExtSer".

Ricordiamo che il casting è un'operazione, consentita dal linguaggio "C", che permette di convertire il tipo di un dato, in qualsiasi punto del programma. La conversione avviene solamente per il compilatore: il dato su cui viene operato il "cast" non viene trasformato effettivamente! Supponiamo di avere la variabile "dec" definita come float e la variabile "num" definita come int. E' chiaro che non possiamo eseguire direttamente operazioni tra le due variabili, in quanto il tipo dei dati è diverso. Nel caso in cui dovessimo memorizzare la parte intera della variabile "dec" all'interno della variabile "num", potremmo fare ricorso ad un cast:

```
num = (int)dec;
```

Il contenuto della variabile "dec" verrà convertito in un numero intero grazie all'operatore di cast "(int)" e quindi assegnato alla variabile "num". In ogni caso (e non è possibile fare diversamente), "dec" rimarrà, per tutta l'esecuzione del programma, una variabile di tipo "float". In Amiga l'uso dell'operatore cast è molto frequente e permette di utilizzare la stessa funzione con dati di tipo diverso (nei limiti del possibile).

Chiusa questa breve parentesi sul cast, torniamo al nostro esempio. Nella funzione CreateExtIO abbiamo specificato come porta "MyPort", cioè quella creata in precedenza per questo scopo. Ora che abbiamo sia la porta che la struttura dati su cui lavorare (controllate sempre che questo sia vero con delle istruzioni "if"), possiamo finalmente aprire il dispositivo seriale:

```
OpenDevice("serial.device",0,SerialData,0);
```

La funzione di Exec OpenDevice( ) viene utilizzata per aprire un generico dispositivo. Questa funzione necessita di quattro parametri: il nome del dispositivo da aprire, il numero di unità, la struttura di dati da usare ed eventuali flag.

Il nome, ovviamente, sarà "serial.device", il numero di unità sarà zero, la struttuta dati da usare sarà "SerialData", cioè la struttura che abbiamo creato e inizializzato in precedenza. Il valore relativo ai flag viene lasciato a zero: non ci serve.

Il serial.device può essere utilizzato sia in modo esclusivo che condiviso. L'accesso esclusivo permette l'uso della porta seriale solamente al task che ha aperto il dispositivo, mentre l'accesso condiviso permette l'uso della porta seriale a più task. Per abilitare il modo condiviso, occorre inizializzare il campo SerialData->io_SerFlags con il valore SERF_SHARED prima di aprire il device.

Oltre ad aprire il dispositivo, la funzione OpenDevice( ) provvede anche a inizializzare correttamente la nostra struttura dati "SerialData" secondo le specifiche e la configurazione della porta seriale in uso. Quando il "serial.device" viene aperto, è inizializzato secondo le specifiche indicate nel file "system-configuration", ovvero mediante i parametri impostati in Preferences. E' ovvio che tali valori non soddisfino sempre le nostre esigenze, quindi possiamo programmare il dispositivo seriale, una volta aperto, in modo che assuma la configurazione a noi necessaria (velocità di trasmissione, numero di bit di dati, ecc.).

Il dispositivo seriale può essere pilotato attraverso comandi standard (identici per tutti i dispositivi) e comandi dedicati (utilizzati esclusivamente da questo dispositivo). Per evitare errori noiosi, può essere utile ricordare che i comandi dei dispositivi sono sempre maiuscoli. I comandi devono essere specificati nel campo IOExtSer->IOSer.io_Command. Nel nostro caso, è ovvio che "IOExtSer" dovrà essere sostituito da "SerialData".

I comandi standard più utili per la gestione del dispositivo seriale sono: CMD_WRITE, che scrive sul dispositivo i dati richiesti e CMD_READ, che legge i dati provenienti dalla porta seriale.

Tutti i comandi specifici del dispositivo seriale iniziano con il suffisso "SD", e sono: SDCMD_QUERY, SDCMD_BREAK e SDCMD_SETPARAMS.

SDCMD_QUERY viene utilizzato per conoscere il numero di byte ricevuti dalla porta seriale e attualmente contenuti nel buffer.

SDCMD_BREAK viene utilizzato per inviare un segnale di interruzione alla porta seriale.

SDCMD_SETPARAMS permette di configurare il dispositivo seriale.

Se i valori di default previsti per il dispositivo seriale non sono di nostro gradimento, potremo cambiarli. L'impostazione delle specifiche del dispositivo seriale avviene mediante il comando "SDCMD_SETPARAMS". L'esempio che segue è tratto dal programma "Seriale.c" (che troverete allegato al dischetto di questa rivista) e fa uso del comando "SDCMD_SETPARAMS":

```
SerialData->IOSer.io_Command=SDCMD_SETPARAMS;
SerialData->io_RBufLen=2048L;
SerialData->io_Baud=2400;
SerialData->io_TermArray.TermArray0=0x0d0a0d0a;
SerialData->io_TermArray.TermArray1=0x0d0a0d0a;
SerialData->io_ReadLen=8;
SerialData->io_WriteLen=8;
SerialData->io_StopBits=1;
```

```
SerialData->io_SerFlags=
SERF_EOFMODE|SERF_XDISABLED;
```

Come detto in precedenza, il campo "IOSer.io_Command" viene utilizzato per specificare il comando da trasmettere al dispositivo; in questo caso vogliamo impostare dei nuovi parametri per la porta seriale.

E' possibile anche riservare una zona di memoria per i dati letti dalla seriale attraverso il campo "io_RBufLen". Questo campo specifica la dimensione del buffer di lettura.

Il dispositivo seriale di Amiga è molto versatile e potente: permette di impostare qualsiasi configurazione di trasmissione e soprattutto di velocità. Il dispositivo seriale accetta, in teoria, velocità di trasmissione che vanno da 110 baud a 292000 baud, anche se per velocità maggiori di 19200 baud si potrebbero incontrare problemi durante la lettura dei dati: in questi casi è preferibile lavorare in linguaggio assembler.

Comunque non siamo assolutamente vincolati nell'uso di velocità standard (2400, 4800, 9600, ecc.) e abbiamo la piena libertà di specificare velocità di trasmissione non convenzionali. Nel nostro esempio, la velocità di trasmissione è di 2400 baud, come specificato nel campo "io_Baud".

I due campi successivi ("io_TermArray.TermArray0" e "io_TermArray.TermArray1") vengono utilizzati per specificare i caratteri che indicano la conclusione della ricezione dati. Il numero massimo di caratteri di terminazione che possiamo specificare è otto: nel caso non avessimo bisogno di tutti questi caratteri, siamo comunque costretti a specificarne otto. Ecco il motivo per cui, nel nostro esempio, abbiamo ripetuto più volte gli stessi caratteri (0x0d e 0x0a).

Se questi campi non venissero totalmente riempiti, i caratteri lasciati liberi saranno uguali a zero. Ciò va evitato, perché istruisce di fatto il dispositivo seriale ad accettare come carattere di terminazione il carattere nullo (cosa che è indesiderabile nella maggior parte dei casi) e quindi è preferibile ripetere la serie di caratteri (o parte di essi), fino al completamento dei campi. L'uso di questi due campi non è comunque obbligatorio e ha solamente effetto se viene specificato "SERF_EOFMODE" nel campo dei flag. Il programma riportato nel dischetto utilizza tale funzione, in quanto attende che venga spedito dal computer collegato ad Amiga il carattere di ritorno a capo.

I campi successivi indicano rispettivamente il numero dei bit che compongono il dato in lettura e scrittura, nonché il numero dei bit di stop.

Il campo "io_SerFlags" viene utilizzato per specificare tutte le caratteristiche di connessione e di lavoro, come, per esempio, l'handshaking, la parità e altro ancora. Nel nostro caso, i flag utilizzati sono "SERF_EOFMODE" e "SERF_XDISABLED":

SERF_EOFMODE abilita l'uso dei caratteri di terminazione indicati nei campi "io_TermArray" ed è quindi necessario per il nostro esempio;

SERF_XDISABLED viene utilizzato per disabilitare la procedura di XON/XOFF.

Dopo aver inizializzato i campi della struttura dati, la si dovrà inviare al dispositivo seriale, in modo da rendere effettivi i parametri desiderati.

```
DoIO((struct IORequest *)SerialData);
```

La funzione Exec DoIO( ) viene utilizzata per trasmettere i comandi al dispositivo in uso (nel nostro caso alla porta seriale) e attendere il completamento del comando utilizzato, prima di restituire il controllo al programma. Ponete la vostra attenzione sull'operatore di cast applicato alla funzione DoIO( ): è necessario, in quanto "SerialData" è un puntatore ad una struttura "IOExtSer", mentre DoIO( ) richiede un puntatore a una struttura di tipo "IORequest". L'operatore di cast (come abbiamo già detto in precedenza) sistemerà il tutto e permetterà al compilatore di lavorare correttamente.

Al lavoro

Ora che abbiamo aperto e inizializzato il dispositivo seriale, è giunto il momento di cominciare a "dialogare" con il dispositivo collegato alla nostra porta seriale.

L'esempio riportato sul dischetto legge semplicemente i dati della porta seriale e risponde con dei messaggi. Il programma continuerà questo lavoro fino a quando non riceverà il messaggio "fine". La prima cosa che faremo all'interno del ciclo è una lettura di dati dalla porta seriale: come in qualsiasi altro genere di richiesta, utilizzeremo la struttura "SerialData":

```
SerialData->IOSer.io_Command=CMD_READ;
SerialData->IOSer.io_Length=BUFLEN;
SerialData->IOSer.io_Data=(APTR)buffer;
```

Questa volta, nel campo "IOSer.io_Command" troviamo il valore "CMD_READ", perché si tratta appunto di una richiesta di lettura.

Il campo "IOSer.io_Length" contiene la dimensione massima del buffer dati in cui intendiamo memorizzare i valori letti.

Il campo "IOSer.io_Data" contiene l'indirizzo relativo alla zona di memoria in cui inizia il buffer dati.

Dopo aver indicato correttamente i valori necessari per la lettura, non ci resta altro da fare che inviare la richiesta al

dispositivo mediante la funzione SendIO( ):

```
SendIO((struct IORequest *)SerialData);
```

In realtà, potremmo utilizzare sia la funzione DoIO( ) che SendIO( ): entrambe permettono di inviare i nostri comandi al dispositivo, ma esiste una differenza fondamentale tra le due, che vale la pena di esaminare. La funzione SendIO( ), a differenza di DoIO( ), non attende che il comando inviato sia stato soddisfatto, ma semplicemente accoda la richiesta al dispositivo e restituisce il controllo al nostro programma. Questo risulta comodo in molti casi, soprattutto quando abbiamo diversi compiti da svolgere contemporaneamente nel nostro programma.

Nel nostro caso, accoderemo la richiesta al dispositivo e attenderemo il suo termine mediante la funzione "Wait". Avremmo potuto utilizzare tranquillamente la funzione DoIO( ) al posto di SendIO( ), ma ho preferito, per puri scopi didattici, utilizzare quest'ultima. Dopo aver accodato la richiesta, dovremo attendere che qualche messaggio arrivi alla porta seriale. Quando questo accadrà, verremo avvisati mediante la porta che abbiamo riservato al dispositivo seriale; dunque, non ci resta che aspettare:

```
Wait(1L<<MyPort->mp_SigBit);
```

La funzione "Wait" risolve in maniera egregia tale compito e, soprattutto, non consuma prezioso tempo macchina, che potrebbe risultare utile ad altri task (Amiga è un computer che opera in multitasking!).
Ponete attenzione all'argomento passato alla funzione: il contenuto di "MyPort->mp_SigBit" rappresenta il numero del bit del segnale che viene utilizzato dalla porta, mentre la funzione "Wait" necessita della maschera relativa a quel determinato bit. L'operatore di shift "<<" risolve, appunto, questo problema, trasformando il bit di segnalazione nella relativa maschera.

Ricordate, comunque, che il dispositivo seriale è stato da noi impostato per leggere i dati fino a quando non incontra uno dei caratteri specificati nel campo "io_TermArray", oppure fino a quando i dati ricevuti avranno superato la dimensione del buffer. La funzione "Wait" attenderà, dunque, fino a che uno di questi due eventi non sia stato soddisfatto.

Passiamo ora all'analisi della parte di codice che si occupa della scrittura sulla porta seriale:

```
SerialData->IOSer.io_Command=CMD_WRITE;
SerialData->IOSer.io_Length=-1;
SerialData->IOSer.io_Data=(APTR)string;
```

La struttura dati ha assunto un'aspetto leggermente diverso.

Il campo "IOSer.io_Command" è stato impostato con il valore "CMD_WRITE" in quanto desideriamo scrivere sul dispositivo. Il campo "IOSer.io_Length" contiene il numero di byte da trasmettere alla porta seriale. Può sembrare strano il valore "-1", ma ha un significato ben preciso: significa che dovranno essere trasmessi tutti i byte a partire dall'indirizzo fornito fino al primo carattere nullo.

Il campo "IOSer.io_Data" contiene l'indirizzo di memoria in cui risiedono i dati da trasmettere. Siccome nel nostro esempio utilizziamo un'array di caratteri (cioè una stringa), ecco che si spiega il valore "-1" nel campo precedente.

Per rendere esecutiva questa richiesta, inviamo al dispositivo la nostra struttura dati mediante la funzione DoIO( ) o SendIO( ). Quando avremo terminato di usare il dispositivo seriale, sarà necessario (anzi, obbligatorio!) rilasciare tutte le risorse che abbiamo utilizzato, compreso il "serial.device".

```
CloseDevice((struct IORequest *)SerialData);
DeleteExtIO((struct IORequest *)SerialData);
DeletePort(MyPort);
```

Le funzioni DeletePort( ), DeleteExtIO( ) e CloseDevice( ) provvedono appunto a quest'ultimo compito.

Il programma incluso nel dischetto contiene un esempio completo. Per farlo funzionare, è necessario collegare Amiga a un altro computer fornito di porta seriale RS-232C, unitamente a un programma di comunicazione.

Dopo averlo mandato in esecuzione, sarà possibile inviare, dal computer collegato ad Amiga, alcuni "comandi", ai quali Amiga risponderà con messaggi appropriati. I "comandi" riconosciuti dal programma sono: host, baud, data, stop, buffer, fine e magic.

Amiga Hardware Reference Manual

La terza edizione

Alberto Geneletti

Dopo l'AmigaDOS, l'Amiga User Interface Style Guide e l'Includes & Autodocs, questo mese abbiamo esaminato attentamente il volume relativo all'hardware dei nuovi sistemi Amiga, disponibile nelle librerie italiane già da alcuni mesi.

La differenza sostanziale rispetto agli altri volumi recensiti nei numeri scorsi riguarda il pubblico al quale questo volume si rivolge. Mentre il ROM Kernel Manual e la Style Guide si rivolgono al programmatore di applicazioni ad alto livello, in particolare allo sviluppo di software interattivo basato su un'interfaccia grafica standard, il volume sotto esame in queste pagine è interamente dedicato al programmatore assembler di routine che interagiscono direttamente con l'hardware: ai cosiddetti "Metal Basher", tanto per usare un termine familiare ai nostri lettori.

Si tratta in pratica di una soluzione vivamente sconsigliata nella prefazione di tutti i manuali dedicati alla programmazione dei sistemi Amiga (e anche l'Amiga Hardware Reference Manual non fa eccezione), poiché soltanto l'interfaccia sistema-utente fornita dalle funzioni di libreria del sistema operativo è in grado di garantire il corretto funzionamento del codice sulle future release del sistema operativo. Nessuna garanzia ufficiale è stata invece rilasciata sulla continuità delle locazioni dei registri hardware, anche se le nuove macchine in produzione hanno mantenuto, saggiamente, locazioni analoghe per i registri di I/O.

Questo libro si rivolge allora a tutti coloro che sono disposti a correre questi rischi, e in particolare agli sviluppatori di videogame, un campo nel quale si desidera un'ottimizzazione molto spinta dell'interazione tra codice e macchina, in modo da valorizzare al massimo le prestazioni messe a disposizione dall'hardware decisamente sofisticato e potente di tutti i modelli Amiga. Altri potenziali utenti di questo manuale sono i progettisti di nuove periferiche che, nello sviluppare i driver per i propri dispositivi fisici, hanno bisogno di sapere come avviene lo scambio di segnali tra Amiga e il mondo esterno tramite i connettori della macchina e le facilitazioni messe a disposizione dal protocollo di autoconfigurazione. Infine, potremmo consigliare questo manuale anche agli utenti e ai programmatori un po' meno smaliziati, perché, se è vero che per guidare una macchina non è necessario conoscere alla perfezione cosa avvenga istante per istante all'interno del cofano, è altrettanto vero che non è possibile andare molto lontano ignorando completamente l'esistenza di pistoni e candele.

Il contenuto

Il libro può essere suddiviso in due parti fondamentali; la prima, che si identifica con i primi otto capitoli, potrebbe essere definita teorica, poiché descrive discorsivamente il funzionamento della macchina; la seconda invece, costituita dalle appendici, svolge essenzialmente una funzione di riferimento, essendo composta da tabelle di rapida consultazione, che risultano molto utili, se non indispensabili, per la programmazione dei registri hardware.

La teoria

Il primo capitolo costituisce un'ampia introduzione destinata al lettore meno esperto, che descrive l'architettura generale del sistema da un punto di vista funzionale, senza scendere nel dettaglio. Vengono inoltre elencate tutte le prestazioni implementate sui nuovi modelli, in particolare sul 3000, per quanto riguarda la possibilità di indirizzare memoria di vario tipo, di interfacciare segnali video e dispositivi con interfaccia SCSI.

Inoltre, vengono fornite direttive generali sulla programmazione in assembler; in particolare viene affrontato il tema del refresh della cache memory da parte di codice automodificante, la presenza nel set del 68000 di istruzioni incompatibili con la struttura del bus condiviso tra CPU e chip custom, il diverso utilizzo del'8520 da parte del 2.0, le procedure di chiamata delle system call da assembler e, infine, vengono dati alcuni chiarimenti di carattere generale sugli esempi presenti nel volume.

A questo ultimo proposito ci duole rilevare che il libro si dimostra piuttosto carente: i sorgenti dimostrativi sono infatti ridotti al minimo, quasi a sottolineare il fatto che, a questo livello, è piuttosto difficile produrre codice in grado di offrire sufficienti garanzie di compatibilità per il futuro. L'utente della manualistica ufficiale dovrebbe del resto essere abituato a questo atteggiamento conciso e criptico, dal momento che vengono proposte, come abbiamo avuto modo di sottolineare in altre occasioni, spiegazioni quasi

sempre complete, ma che richiedono un certo numero di tentativi prima di poter essere comprese completamente.

Se ciò può non creare particolari problemi nella programmazione ad alto livello, l'esperienza dimostra che lavorando direttamente a contatto con l'hardware, è facile arrivare a conclusioni sbagliate, sviluppando codice che funziona solo in determinate situazioni. Per questo, riteniamo che qualche listato in più, soprattutto per quanto riguarda la gestione delle unità disco, della seriale e della parallela, o magari un po' più completo, ad esempio nel caso della creazione delle copper list, sarebbe sicuramente risultato una gradita novità per questa nuova edizione. Come se non bastasse, questi listati sono stati "snelliti" definendo un improbabile file include hardware/examples.i, la cui definizione completa è riportata nell'appendice I.

Questo file ridefinisce, usando le maiuscole, gli identificatori dei registri già definiti nel file di inclusione hardware/custom.i, perdendo a volte qualche lettera; crea due identificatori analoghi per le locazioni a 32 bit, uno per la word più significativa e l'altro per quella meno significativa; usa per i registri colore, e per i bitplane, degli array dei quali viene fornita in hardware/custom.i soltanto la base.

Il problema pratico consiste nel fatto che questo file include non è presente nell'apposita di directory di nessun assemblatore in circolazione, non trattandosi di un include ufficiale, a differenza di hardware/custom.i.
Per questo motivo, desiderando assemblare uno qualsiasi dei sorgenti esemplificativi, ci si trova costretti o a digitare una volta per tutte questo file di inclusione aggiuntivo, oppure ad effettuare conversioni non sempre immediate con la simbologia di hardware/custom.i; si tratta in conclusione di una scelta didattica decisamente controproducente.

I capitoli seguenti sono dedicati invece ad argomenti specifici e in particolare alla programmazione del Copper per la gestione del refresh del video, alla costruzione dei playfield, alla visualizzazione degli sprite, alla descrizione dei quattro canali audio, alla programmazione del Blitter. Gli ultimi due capitoli integrano, infine, la descrizione del funzionamento delle singole parti della macchina, affrontando argomenti importantissimi, quali le interruzioni, la gestione del DMA, il processo di autoboot, l'interfaccia dei floppy, delle porte giochi, della seriale e della parallela.

Come abbiamo già affermato, proprio questa parte del libro, che tratta argomenti di complessità decisamente superiore a quella dei capitoli precedenti, si rivela decisamente carente dal punto di vista dei sorgenti esemplificativi. Dal momento che ogni capitolo tratta un argomento diverso, sembrerebbe a prima vista possibile rivolgere la propria attenzione a un aspetto qualsiasi, rinviando la lettura degli altri capitoli a un secondo momento; l'esperienza ci ha dimostrato che questo non è assolutamente possibile.

Sebbene gli argomenti vengano presentati in modo abbastanza indipendente, al momento di passare alla pratica si scopre che tutto ciò che viene detto in un solo capitolo non funziona autonomamente; questo perché le varie parti della macchina interagiscono continuamente tra di loro, procurando quelli che il neoprogrammatore potrebbe considerare "fastidi" apparentemente ineliminabili, ma che forniscono spesso una sinergia ineguagliabile di potenza al programmatore esperto. Fate attenzione in particolare a non sottovalutare il Copper; scoprirete allora insignificanti note del tipo "...questi registri vengono scritti esclusivamente dalla copper list..." praticamente in ogni capitolo.

La chiave di lettura che vi proponiamo, e che dovrebbe essere evidenziata nella prefazione del libro stesso, è quindi strettamente sequenziale; in ogni caso non si può pensare di costruire un playfield senza prima conoscere tutto il resto. Il discorso didattico nel suo complesso è comunque molto ben costruito, corredato da spiegazioni esaurienti e facilmente accessibili anche al lettore meno esperto; l'impressione è tuttavia che ci si dilunghi un po' troppo sugli aspetti più semplici, e che non si desideri appesantire la trattazione con argomenti troppo complessi.

Le appendici

Di carattere decisamente tecnico è invece la seconda parte del libro, costituita dalle appendici, che costituiscono il manuale di riferimento vero e proprio. In esse troviamo l'elenco completo dei registri custom, tanto in ordine alfabetico quanto di indirizzo, le mappe di memoria delle varie configurazioni, la descrizione dei segnali presenti ai connettori e sui piedini dei vari chip, la descrizione completa del funzionamento delle CIA 8520 e dell'interfaccia della tastiera.

Particolare attenzione merita, in questa terza edizione, l'appendice C, dedicata interamente alla descrizione delle nuove caratteristiche e dei nuovi registri messi a disposizione dall'Enhanced Chip Set (ECS), la versione dei chip custom presenti sui nuovi Amiga dotati del sistema operativo 2.0, che possono essere considerati i veri protagonisti di questa edizione. In questa appendice è possibile trovare informazioni che vanno dal funzionamento delle nuove risoluzioni video, alla gestione delle nuove frequenze di scansione, alle estensioni nel supporto del Genlock, all'ampliamento della capacità del Blitter, e di alcuni registri preesistenti dei quali è stato ampliato il numero di bit significativi.

Tutte queste informazioni sono state riunite saggiamente in quest'unica appendice, in modo da rendere ben chiara la distinzione tra i vecchi e i nuovi modelli. Un'altra sezione particolarmente interessante per i progettisti di nuove schede e nuove periferiche è senza dubbio l'appendice dedicata alla descrizione del bus di espansione Zorro III a 32 bit, implementato su Amiga 3000. In questa sede viene analizzata la compatibilità con Zorro II (il bus standard del

2000), descrivendo ampiamente lo scambio di segnali sul bus di controllo nell'ambito delle due differenti architetture. Vengono, inoltre, fornite specifiche meccaniche circa l'ingombro delle schede Zorro III compatibili e, infine, viene spiegato l'importante meccanismo di autoconfigurazione.

La terza edizione

Tra tutti i nuovi manuali di questa terza edizione della documentazione ufficiale, l'Amiga Hardware Reference Manual è senza dubbio quello che ha subito il minor numero di modifiche rispetto all'edizione precedente. Tutta la parte teorica è rimasta assolutamente invariata; questo perché le caratteristiche e la programmazione dei nuovi registri implementati nell'ECS non vengono trattati direttamente nei capitoli: questi conservano così gli stessi paragrafi e lo stesso limitato numero di esempi, e non sono stati riscritti; ogni volta che la trattazione si avvicina ad una delle nuove caratteristica viene semplicemente inserita una nota che rimanda all'appendice C.

Anche certe appendici sono rimaste pressoché invariate, mentre sono state introdotte le due importanti sezioni su ECS e Zorro III delle quali abbiamo già parlato; qualche novità di rilievo presenta l'appendice dedicata alle mappe di memoria, nella quale è possibile trovare ora anche quella del 3000. E' evidente quindi l'intenzione di mantenere intatto il contenuto della seconda edizione e di aggiungere tutto il materiale che costituisce l'upgrade per ECS e Amiga 3000 in alcune zone circoscritte e ben definite.

Tutto ciò fa pensare ancora una volta all'inadeguatezza della distribuzione dell'upgrade delle informazioni da parte di Addison Wesley. E' infatti assurdo che l'utente possessore della seconda edizione sia costretto a pagare nuovamente il prezzo dell'intera maualistica per ritrovarsi poi in possesso di un volume che ripete per la maggior parte informazioni identiche a quelle già in suo possesso; decisamente più conveniente (ma probabilmente non per la casa editrice) sarebbe la pubblicazione di un opuscolo di una settantina di pagine nel quale venisse riunito esclusivamente il contenuto delle nuove appendici.

Conclusioni

Trattandosi della documentazione ufficiale, anche in questo caso il volume risulta indispensabili al programmatore Amiga; pur essendo dedicato all'hardware, viene infatti affrontato l'aspetto funzionale, proprio quello che interessa lo sviluppatore di software, e non quello strutturale, che riguarda più decisamente il costruttore di hardware o il riparatore.

Potrebbe sembrare strano che un volume dedicato all'hardware non riporti nemmeno il circuito completo della macchina, che, sebbene non documentabile esaurientemente in un'appendice, avrebbe potuto soddisfare la curiosità di qualche lettore. Ma il tecnico hardware ha a disposizione testi più specifici (come l'"A500/A2000 Technical Reference Manual"), riservati tuttavia ai programmatori registrati presso la Commodore. L'Amiga Hardware Reference Manual è invece in grado di soddisfare le esigenze del programmatore assembler, anche se, come abbiamo già detto, non può sostituire completamente l'esperienza.

Amiga Hardware Reference Manual
Third Edition 1991
Addison Wesley — Amiga Technical Reference Series
Lire 65.000 (circa).

(segue da pag. 44)

Architettura dei computer Amiga

Bibliografia

- Commodore Amiga A500/A2000 Technical Reference Manual
Questo libro contiene tutte le specifiche del protocollo Autoconfig, dei bus Zorro e Zorro II, della scheda Janus (XT) e del controller A2090.
E' indispensabile per progettare schede. Un tempo era disponibile solo per gli sviluppatori, ma ora, almeno in Germania e U.S.A., è in commercio.

- Amiga Hardware Reference Manual - terza edizione, Addison Wesley
Edito dalla Addison Wesley, è facilmente reperibile e si interessa perlopiù della programmazione a basso livello del computer.

- G. Kane, Il manuale MC68000, ISBN 88-7700-017-1
Edito dalla McGRAW-HILL, è in italiano e riassume brevemente le principali caratteristiche del 68000 e 68010.

- Alan Clements, Microprocessor Interfacing and the 68000 Peripherals and Systems, ISBN 0-471-91575-0
Edito dalla John Wiley & Sons, descrive in dettaglio svariati interfacciamenti del 68000 e i principali problemi e soluzioni.

Innovativo, più flessibile, più potente

Anche in Italia finalmente

MINIX 1.5

MILANO - Minix 1.5 è un sistema operativo nuovo, flessibile, potente. Sviluppato a scopo didattico è oggi un vero e proprio software di base che rispetta tutte le caratteristiche di UNIX. La repentina espansione del sistema operativo UNIX nell'ultimo decenn...

con un proprio codice sorgente, modificabile per poterne personalizzare l'utilizzo. Ideato e scritto dal Professor Andrew S. Tanenbaum e prodotto da Prentice Hall è distribuito in Italia dal Gruppo Editoriale Jackson leader indiscusso nel campo della ...matica e dell'elettronica ...azioni

Tutte le caratteristiche:
• MULTIPROGRAMMAZIONE E MULTIUTENZA • EDITOR: ed, ex, vi, emacs • OLTRE 175 UTILITY: cat, grep, kermit, make... • OLTRE 200 FUNZIONI DI LIBRERIA: atoi, fork, malloc...• SUPPORTO RS-232 • CODICE SORGENTE C SU DISCO • COMPILATORE C (Kernighan Ritchie)

MINIX 1.5

Distribuito nelle migliori librerie e nei computershop

versione PC IBM 3 ½"
Cod.DMX3 12 dischi L.275.000

versione PC IBM 5 ¼"
Cod.DMX5 17 floppy L.275.000

versione Macintosh
Cod.DMXM 8 dischi L.275.000

versione AMIGA
Cod.DMXA 9 dischi L.275.000

Da spedire in busta chiusa a: **GRUPPO EDITORIALE JACKSON**, Via Rosellini 12 - 20124 Milano

Sì, inviatemi i volumi sottoelencati

INDICARE CHIARAMENTE CODICI E QUANTITÀ DEI VOLUMI RICHIESTI

Codice	Q.tà	Codice	Q.tà	Codice	Q.tà	Codice	Q.tà	Codice	Q.tà	Codice	Q.tà

L. 6.000 per contributo fisso spese di spedizione

❑ Sono titolare della Jackson Card '92 n° ______ e ho diritto allo sconto del 10% (fino al 31/12/92) ❑ Non sono titolare

Modalità di pagamento:
❑ Allego assegno n° ______ Banca ______ di L. ______
❑ Contro Assegno postale al ricevimento dei volumi
❑ Richiedo fattura (Partita IVA n° ______)

Cognome e nome ______
Via ______ n° ______
Cap ______ Città ______ Prov. ______
Tel. ( ) ______
Firma ______
Data ______

MINIX 1.5 è distribuito da

menu dai quali è semplice ottenere immediatamente una prima elaborazione a forma di rosa. Selezionando "Choices/Watch Evolution..." si otterrà un primo esempio, ma per personalizzare il risultato è necessario modificare le opzioni "Choices/Change n & d values" che modifica i valori dell'equazione polare r=sin(n*d) utilizzata per disegnare le rose.
Per ottenere degli effetti particolari, può essere necessario variare il parametro del numero di suddivisioni del cerchio con il menu "Choices/Change circle subdivisions" oppure il tipo di algoritmo utilizzato per il tracciamento con il menu "Type/Variation 1" oppure "Type/Variation 2".
Nel caso non riuscite ad entrare nell'ottica puramente matematica del programma, divertitevi a sperimentare e nel caso realizzate qualcosa di veramente memorabile usate l'utile funzione di salvataggio su memoria di massa per lasciare un segno ai posteri.
Attenzione: questo programma non funziona su modelli di Amiga con sistema operativo superiore alla release 1.3

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

UTILIZZO
Workbench: Doppio Click sull'icona

FILE DI SUPPORTO
Nessuno.

TURBOdisk

Frank Mariak

TURBOdisk è uno di quei programmi che funzionano al meglio se eseguiti nella Startup-Sequence del proprio disco di lavoro. TURBOdisk è la risposta alla proverbiale "lentezza" dei drive gestiti da AmigaDOS, in quanto si occupa di ridurre il più possibile i tempi di attesa. Come?
Tecnicamente intercettando la trackdisk.device e allocando dei buffer di memoria cache per ogni drive "visto" dal sistema.
In parole semplici vuol dire che TURBOdisk, una volta eseguito dalla finestra CLI, si installa automaticamente dal sistema scomparendo dalla vista dell'utente.
Da quel momento in poi agirà nell'ombra, velocizzando l'accesso ai drive senza utilizzare memoria necessaria al sistema, nel caso ce ne fosse bisogno.
Nel caso fosse necessario rimuoverlo dalla memoria, per una qualunque ragione, sarà sufficiente eseguire nuovamente TURBOdisk da una finestra CLI.
Un appropriato messaggio confermerà l'avvenuta rimozione del programma dal sistema. TURBOdisk è un programmino piccolo, efficente e sicuro.
Inseritelo tranquillamente nell'elenco dei programmi da eseguire nella Startup-Sequence, non ve ne pentirete!

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3/2.0

UTILIZZO
CLI: CD TURBOdisk [enter
TURBOdisk [enter

FILE DI SUPPORTO
Nessuno.

Twin Express v1.1

OMNICODE

Questo mese è dedicato ai possessori di Amiga che danno uno sguardo insistente anche al mondo PC (MS-DOS compatibili).
Oltre alla versione dimostrativa di PC-Task in questo numero di Amiga Magazine trovate Twin Express, un package professionale per il trasferimento di dati via null-modem. Un null-modem è un'interfaccia RS-232 (seriale) che permette di collegare via cavo due computer collegati tramite la stessa porta di comunicazione.
Twin Express utilizza al massimo delle possibilità la porta seriale di Amiga senza peraltro utilizzare la serial.device ottenendo la stratosferica velocità media di trasferimento di 22.000 byte/secondo (veramente niente male!).
Il package di Twin Express è composto da un file eseguibile per i computer Amiga chiamato "Twin", un file per i computer MS-DOS chiamato "TWINPC.EXE", un manuale di 30 pagine chiamato "TwinDoc" e dei file informativi.
Tutti i file si trovano nella directory TwinExpress del dischetto Amiga-Magazine.
Ammettiamo di voler effettuare un trasferimento di dati tra due Amiga collegati via null-modem utilizzando questo programma.
Innanziutto è necessario eseguire il programma cliccando sulla sua icona.
Quello che ci verrà presentato è un menu dal quale è possibile effettuare tutta una serie di comandi di trasferimento e di controllo. Per trasferire un file da un Amiga all'altro basta digitare il comando "COPY FILE.TXT ~FILE.TXT".
Il simbolo di tilde (~) indica il computer collegato in remote, quindi tutti i comandi che devono agire sul computer a distanza devono specificare il simbolo di tilde. Di conseguenza se si vuole fare una directory del computer in remote basta digitare "DIR ~.TXT".
Qualcuno potrebbe essere rimasto sconcertato dagli esempi riportati. Bene, fortunatamente all'interno del programma TwinExpress è possibile ottenere informazioni sui comandi disponibili con "HELP COMMANDS", o alcuni esempi di utilizzo con "HELP EXAMPLES" oppure ancora utili informazioni su come realizzare un null-modem con "HELP CABLE".

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3/2.0

UTILIZZO
Workbench: doppio click sull'icona

FILE DI SUPPORTO
Per il trasferimento da o verso un PC, è necessario il file TWINPC.EXE che si trova nella directory TwinExpress del disco AmigaMagazine.

View v3.0

Michael W. Hartman

View è un programma per visualizzare file ILBM e ANIM, oltre che file DCTV. Quest'ultima opzione è ovviamente utile solo nel caso si disponga dell'hardware DCTV, mentre la possibilità di vedere immagini e animazioni è spesso indispensabile durante le normali fasi di utilizzo di Amiga.
View è un visualizzatore della generazione più recente, non si limita quindi a svolgere il suo compito, ma offre un'interfaccia utente di prima categoria, caratteristica dei sistemi operativi più evoluti.
View può essere usato in ben tre modi diversi. Il primo e più immediato (almeno per noi) è quello di "passare" come argomento al programma il file che si intende visualizzare, il secondo è settando come Default Tool dell'icona associata al file che si intende visualizzare il programma View, il terzo e ultimo è quello di eseguire il programma View e utilizzare il file requester interno.
Nel primo e nel secondo caso il programma View visualizza il file passato come parametro, attende la pressione del tasto sinistro del mouse ed esce, mentre nel terzo il programma visualizza il file, attende la pressione del tasto del mouse e ritorna al file requester di partenza.
In qualunque modo venga utilizzato, View visualizza utili informazioni a proposito del file che si sta per visualizzare come la risoluzione, il tipo, la grandezza e il nome, oltre ad offrire la possibilità di controllo sulla velocità di visualizzazione delle animazioni.
View merita senz'altro di stare nell'elenco delle utility indispensabili all'utente Amiga che si rispetti o che comunque voglia un solo programma per visualizzare, ad esempio, i propri risultati artistici conseguiti con DeluxePaint.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3/2.0

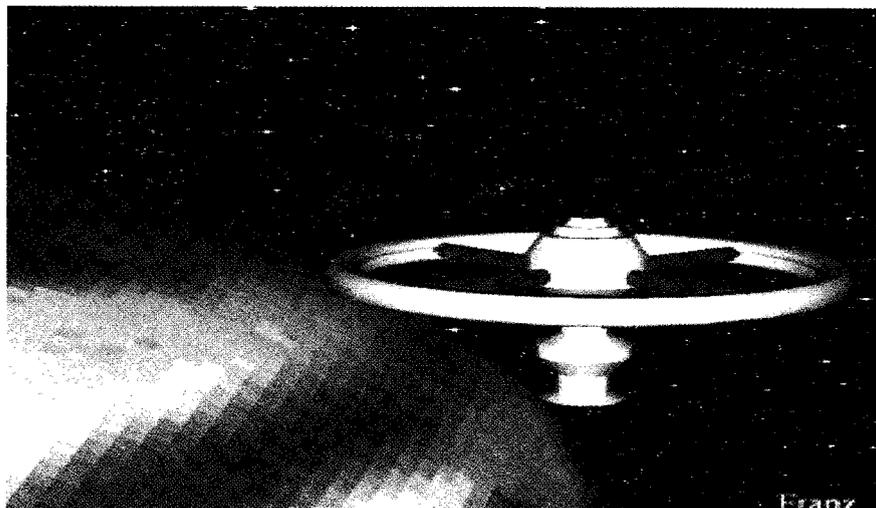
UTILIZZO
Workbench: doppio click sull'icona dopo aver settato il Default Tool
CLI: CD View3.0 [enter
View "Nome_File_Immagine_o_Animazione" [enter - modo diretto
oppure
View [enter - modo interattivo

FILE DI SUPPORTO
Nella directory View 3.0 sono presenti due belle immagini realizzate con pacchetti di Ray Tracing per Amiga: "Penoc.iff" e "Space.ham".

NOTE

Ricordiamo che per motivi di spazio le directory C, DEVS, L e LIBS del disco di Amiga Magazine non sono complete come dovrebbero essere, soprattutto per poter funzionare come dischi con cui effettuare il boot con il KickStart v2.04. Si consiglia quindi ai possessori di Amiga 500 Plus che dovessero riscontrare dei problemi di funzionamento di effettuare il boot dal proprio disco Workbench e, successivamente, sostituire nel drive il disco con quello di Amiga Magazine.
I comandi Dir e List del CLI sono stati sostituiti con il più compatto ed efficente LS, quindi usate quest'ultimo per visualizzare il contenuto di una directory.
Le librerie presenti nella directory LIBS del disco Amiga Magazine arp.library, req.library, reqtools.library, explode.library e powerpacker.library sono di pubblico dominio e quindi liberamente distribuibili. Vi consigliamo di copiarle sul vostro disco di lavoro (o harddisk) perché sono indispensabili al funzionamento di molti dei programmi di pubblico dominio presenti sul disco allegato ad Amiga Magazine ogni mese.

Attenzione

Quando viene indicata la scritta tra parentesi quadre come [enter, significa che dovete premere il tasto con scritto Enter presente sulla destra del tastierino numerico oppure il tasto di ritorno carrello posto sulla sinistra dei tasti cursore.

AMIGA 600

a cura della redazione

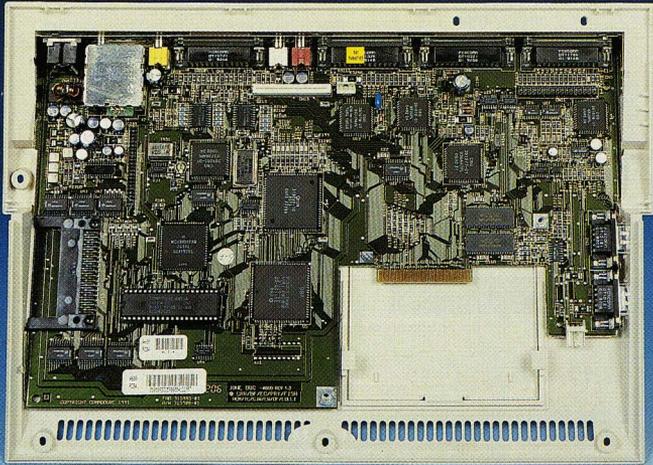
Uno sguardo ravvicinato.

Dopo le notizie raccolte al CeBIT tedesco, abbiamo potuto provare direttamente Amiga 600 presso la Commodore Italiana, che ha iniziato la distribuzione del prodotto ai rivenditori il 2 Aprile.
La prova non ha, di fatto, aggiunto molto a quanto dicevamo la scorsa volta sul nuovo modello della Commodore, ma abbiamo potuto aprire il computer, osservare da vicino il software di sistema e trarre alcune iniziali impressioni d'uso. Ricordiamo che i modelli disponibili sono due: Amiga 600 (quello provato) e Amiga 600HD (nelle foto), il secondo è dotato di hard disk interno da 2.5" di tipo IDE.

L'ASPETTO ESTERNO

Amiga 600 è molto compatto e leggero. Occupa, praticamente, la metà dello spazio di un 500 e trova posto comodamente su qualsiasi scrivania. Il colore classico della linea Amiga è cambiato: ora tende nettamente al bianco.
La tastiera, al tocco, appare simile a quella del 500, ma l'inclinazione è leggermente più accentuata. Le dimensioni dei tasti sono di fatto le stesse. E' scomparso il tastierino numerico, mentre cursori, Del e Help sono stati addossati alla tastiera principale. Del e Help appaiono anche più piccoli. La tastiera viene prodotta in varie versioni, adatte alle diverse lingue nazionali.

La scomparsa del tastierino numerico potrebbe creare dei problemi al software che richiede la sua presenza per .determinate funzioni. Non sono molti i programmi di questo tipo e di solito esistono delle alternative ai tasti (menu, mouse, gadget), per cui il problema non dovrebbe essere tale da inibire l'uso dei programmi. Potrebbe forse risentirne la giocabilità di alcuni videogiochi. I tre led sulla destra servono per l'alimentazione, il floppy interno e l'hard disk interno.
La parte terminale del 600 presenta un piccolo gradino che collega il piano della tastiera a quello dove giacciono le feritorie: quest'ultimo mantiene la stessa inclinazione.
Un'abitudine diffusa è quella di appoggiare i floppy inutilizzati sulla parte terminale del 500: è ancora possibile farlo sul 600, anche se i floppy ora restano inclinati e sporgono dal fondo della macchina.
Sul fianco destro compare la presa per il mouse e per il joystick; questa posizione costringe a tenere il tappetino del mouse leggermente staccato dalla tastiera, ma essendo le dimensioni di questa molto limitate (se confrontate con quelle del 500), il problema praticamente scompare. Sarebbe

Il 600HD con i tre manuali, il Workbench e il disco di installazione per hard disk.

La scheda madre dell'A600 HD.

comunque stato meglio se la spina del mouse non fosse stata la solita (in gomma piena e piuttosto allungata), per limitare, appunto, la sporgenza dal corpo della macchina. Il mouse è lo stesso in dotazione al 500.
La feritoia per i floppy segue l'inclinazione della tastiera: occorre farci l'abitudine. Il drive appare un po' più rientrato nel corpo della macchina rispetto al 500: ciò implica che bisogna spingere i floppy più a fondo, prima che il drive li "digerisca". Il pulsante per l'espulsione dei drive non sporge come quello dei 500 e appare più morbido al tocco.
Sul retro compaiono tutte le porte standard del 500. Ma c'è qualche piccolo cambiamento. La presa dell'alimentatore è diventata la prima a destra (guardando dal retro): è sicuramente meglio così. L'alimentatore, da 23 Watt come quello del 500, appare anch'esso bianco e molto leggero: il 600 assorbe meno del 500, per cui resta a disposizione una maggiore potenza per le periferiche esterne (forse, adesso, sarà possibile collegare tre floppy esterni senza dover ricorrere ad un alimentatore aggiuntivo).
Subito a sinistra troviamo i connettori video. Sono tre: RGB, videocomposito a colori di ottima qualità e modulato, per il collegamento diretto a una presa d'antenna TV. Assieme al 600 è fornito un lungo cavo d'antenna; è dunque possibile mettere in funzione il 600 non appena lo si estrae dalla scatola, collegandolo al televisore.
La pluralità di porte video è una delle innovazioni più significative del 600: grazie al videocomposito è possibile registrare direttamente su VCR (e con esiti ben superiori a quelli consentiti dal vecchio modulatore A520) il segnale prodotto dal 600; grazie alla presa TV è possibile portare il 600 in giro (in vacanza, dagli amici, dai clienti...), senza trascinarsi dietro anche il monitor. Indubbiamente, la presa TV fa del 600 una significativa alternativa alle console e dovrebbe contribuire ad allargare ulteriormente il mercato Amiga.
Sul lato sinistro compare unicamente il connettore per le RAM Card. E' dunque scomparso il bus d'espansione del 500: il controller A590 e il lettore di CD-ROM, atteso a breve, sono incompatibili con il 600.
Sul fondo compare un cassettino che permette di accedere al connettore per l'espansione di memoria interna. Lo spazio a disposizione è inferiore a quello presente nel 500.

ALL'INTERNO

Per aprire il 600 basta svitare le 4 viti a croce poste sul fondo della macchina. Il cabinet in plastica si divide in due: alla parte superiore resta collegata la tastiera, protetta da una piastra in metallo; all'altra, tutto il resto della macchina.
Sulla griglia di protezione, per i disturbi radio, trovano posto il drive per i floppy e una slitta su cui va appoggiato l'eventuale hard disk interno. Rimosso il drive (un Chinon FZ-354 piuttosto silenzioso, ma non ad alta densità), mediante le tre viti, e la slitta, si può accedere alla piastra, svitando l'unica vite che la collega alla griglia e al cabinet.
La scheda madre, a due strati, sembra deserta: tutti i componenti hanno subito una terrificante "cura dimagrante". Le resistenze sono come dei puntini che costellano il circuito stampato. I chip custom hanno dimensioni ridotte. Anche il 68000 è cambiato. Non ci sono zoccoli se non per la ROM. E' l'effetto della nuova tecnologia SMD adottata dalla Commodore per il 600 e che, ben presto, verrà estesa ad altri modelli della linea Amiga.
I vantaggi di tale tecnologia sono notevoli: abbassamento dei costi di produzione e dei consumi, aumento dell'affidabilità dei circuiti e quindi drastica riduzione della percentuale di guasti. Per riparare tale tipo di schede occorrono speciali attrezzature all'infrarosso, di cui man mano verranno dotati tutti i centri di assistenza Commodore. Si è già provveduto a distribuirle a cinque di questi centri, disseminati sul territorio nazionale.
Ricordiamo che tale tecnologia rende impossibile collegare al 600 tutte le espansioni che vanno montate sullo zoccolo di qualche chip preesistente.
Non è neanche possibile usare sul 600 quelle schedine che permettono di montare due tipi di ROM: in questo caso lo zoccolo esiste, quello che manca è lo spazio fisico per inserire la scheda.
Abbiamo visto due piastre: su quella del 600 compariva la dicitura A300 (era il nome previsto inizialmente per il modello senza hard disk) e il numero di revisione 1.1 su quella del 600HD (nelle foto) compariva la dicitura A600 e il numero di revisione 1.3.
La macchina è dotata, di serie, di 1 Mb di Chip RAM costituita da due chip da 256x16 Kbit.
Le due CIA, irriconoscibili, sono state protette contro l'esterno da partitori resistenza condensatore, che risultano più efficaci delle EMI dei modelli precedenti.
Il Denise (8373) e il Fat Agnus (8375) corrispondono a quelli montati sul 500 Plus (accesso a 2 Mb di Chip e nuovi modi grafici),

viene però a mancare lo switch hardware per passare dal modo NTSC a quello PAL. Ora l'operazione si può effettuare esclusivamente via software.
Il nuovo chip Gayle sostituisce il vecchio Gary e comprende la logica di controllo delle RAM Card e dell'hard disk IDE. Nei pressi del modulatore TV, compare il connettore per Hard Disk di tipo IDE. La scelta dell'IDE (invece dello SCSI) è stata motivata da ragioni di costo, in quanto gli hard disk da 2.5" (montati anche nei laptop) sono molto più economici di quelli SCSI.
Verso la parte anteriore della macchina compare il pettine per l'espansione di memoria: è diverso da quello del 500 (e quindi incompatibile con le schede per tale modello) ed è destinato ad accogliere l'espansione di memoria da 1 Mb della Commodore (verrà chiamata molto probabilmente A601), che porta la memoria Chip a 2 Mb e aggiunge l'orologio con batteria tampone. Non sarà impossibile vedere in futuro altre schede, magari per la memoria Fast, perché sul pettine sono riportati i segnali sia del bus Chip che del bus della CPU. Il connettore per le RAM Card è posto all'estrema sinistra: si tratta di un connettore in standard PCMCIA 2.0; ciò rende la porta compatibile con quelle montate su palmtop dell'ultima generazione, come quelli della Memorex e della HP. La porta montata sul CDTV non è invece compatibile, per ora, con tale standard. Va precisato che ad essa potrebbero essere collegati, in futuro, dispositivi diversi dalle RAM Card: per esempio, schede Ethernet, controller SCSI e così via. Di fatto, va considerato una specie di sostituto del bus d'espansione del 500, capace di raggiungere elevatissime velocità nel trasferimento dei dati (quelle della RAM, appunto), trasferimento che avviene, fra l'altro, a 16 bit.

SCHEDA PRODOTTO

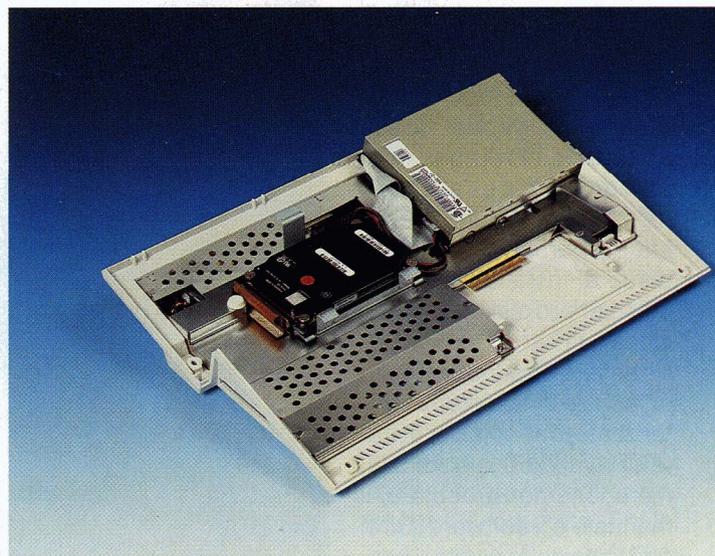
Nome Prodotto: Amiga 600
Casa Prodruttrice: Commodore
Distribuito da: Commodore Italiana - V.le Fulvio Testi 280 - Milano - Tel. 02/661231
Prezzo: Lire 669.000 + IVA
Giudizio: ottimo
Pro: dimensioni, uscite video.
Contro: poco espandibile.
Configurazione della prova: A600

MANUALI

I manuali del 600, tutti in italiano, sono due: il primo è una breve introduzione al 600, con molte immagini. Manca però lo schema elettrico, presente invece sul volume corrispondente del 500. Fra l'altro, si dice, erroneamente, che il 600 ha 512K di RAM.
Il secondo è, fondamentalmente, lo stesso che viene fornito con il 500 Plus e riguarda il Workbench 2.0. Ricordiamo che mancano le spiegazioni relative al linguaggio ARexx, fornito di serie con il Kickstart 2.0.
Le informazioni sul controller per hard disk e sul software di installazione si trovano in un terzo manuale, fornito solo con il 600HD.

IL SOFTWARE

Nulla di particolarmente nuovo su questo fronte. La macchina monta le ROM del Kickstart 37.300 (2.05).
I dischetti forniti sono i soliti Workbench, Extras e Font. La release del Workbench è chiamata 2.05 (2.04 era quella del 500 Plus) e corrisponde al numero di versione 37.71 (37.67 sul 500 Plus). Nel 600HD viene fornito un disco aggiuntivo per l'installazione dell'hard disk.
Le differenze del software di sistema rispetto alla versione del 500 Plus sono marginali: è stato introdotto il supporto per l'hard disk e per le RAM Card, oltre a piccole correzioni che rendono il 600 leggermente più compatibile del 500 Plus con il software esistente.
Le RAM Card appaiono al sistema come un handler automount del DOS chiamato CC0:. Quando la card non è inserita, il device appare vuoto: la card corrisponde, da questo punto di vista, a una specie di floppy, che può essere inserito ed estratto dall'utente.
Dal punto di vista software, le RAM Card vengono gestite da nuovi moduli residenti chiamati card.resource e carddisk.device.
A livello utente è possibile modificare la configurazione delle RAM Card, utilizzando il programma PrepCard, fornito nel disco Extras. Due sono le opzioni fondamentali disponibili: la prima fa vedere la memoria della RAM Card come un disco, la seconda aggiunge la memoria direttamente alla Fast di sistema. La RAM Card può dunque costituire un'autentica espansione di memoria Fast. Ci sono però dei limiti a questo livello. Un submenu di PrepCard, che serve, diciamo, per una sorta di "formattazione" a basso livello della Card, offre le seguenti opzioni: RAM statica o RAM dinamica, una velocità di accesso compresa tra i 150 e i 250

L'interno dell'A600 HD: si noti il floppy da 3.5" e l'hard disk da 2.5, 20 Mb.

nanosecondi, una estensione massima di 2 Mb per ogni unità (dette Page), 32 unità al massimo. per un totale teorico di 64 Mb. Se ne deduce che i 64 Mb teorici sono accessibili solo a pagine di 2 Mb. Per ora sono previste dalla Commodore RAM Card da 2 e 4 Mb.
E' possibile anche scegliere il metodo di controllo dell'integrità dei dati presenti nella memoria della Card (CRC o Checksum oppure nessuno). Ricordiamo che le RAM Card possono essere dotate di batteria tampone per la salvaguardia dei dati. Infine, nel caso si scelga di usare la scheda come disco, si possono determinare settori, tracce, cilindri, come in una qualsiasi Mountlist. Chiaramente, la disponibilità di tali opzioni dipende dalla natura della RAM Card aggiunta al sistema. Ricordiamo che potrebbe essere costituita da ROM, che risulta, ovviamente, immodificabile dall'utente. Per quanto riguarda la partizione e la formattazione dell'hard disk non abbiamo trovato un programma specifico nell'A600. Viene invece fornito insieme ad un manuale aggiuntivo a chi acquista direttamente il 600HD. Rimane comunque il problema di recuperare programma a coloro che acquistano il modello base e poi decidono di aggiungere l'hard disk in un secondo momento. La Commodore sta preparando un kit con disco di installazione, manuale e hard disk. L'installazione dell'hard disk dovrebbe essere effettuata presso un centro di assistenza autorizzato, non solo per evitare di invalidare la garanzia, ma anche oer orocurarsi il manuale e il software. Direi che tale scelta risulta perlomeno discutibile: il software di installazione e la manualistica dipendono essenzialmente dal controller, non dalla meccanica dell'hard disk. Chi compra il 600, compra anche il controller, e dovrebbe dunque avere a disposizione tutte le informazioni che gli servono per installare una meccanica di propria scelta. L'hard disk montato sul 600HD, secondo le ultime informazioni, è da 20 Mb e ha un tempo di accesso di 23 ms (una discreta velocità).

I connettori esterni del 600HD.

CONCLUSIONI

Per quanto riguarda la compatibilità con il software esistente, non crediamo che esistano particolari problemi: da questo punto di vista, se un programma funziona sul 500 Plus, dovrebbe funzionare anche sul 600. Sembra addirittura che alcuni giochi che non funzionano sul Plus, girino correttamente sul 600, forse proprio a causa delle piccole correzioni apportate al Kickstart. Per quanto riguarda la destinazione commerciale del 600: Aormai chiaro che l'obiettivo A quello di creare il vero erede del C64 (destinato ad uscire fuori di produzione), cioè l'home computer degli anni '90, il primo dotato di HD interno di serie. La scelta fra 500 Plus e 600 appare per certi versi ardua. Il 500 ha dalla sua il tastierino numerico e una maggiore espandibilità, che significa controller SCSI, lettore di CD-ROM, schede e schedine di oani tipo da collegare internamente (compresi i 68030 e gli emulatori MS-DOS). A favore del 600 giocano la tecnologia SMD, l'hard disk interno, le dimensioni, le RAM Card. l'uscita video modulatae in videocomposito E' probabile che il 600 non solo allarghi ma anche eroda profondamente il mercato del 500, specie se il prezzo finale della versione con hard disk risultasse particolarmente conveniente. Ci sembra che solo la possibilità di rendere il 500 Plus una specie di CDTV, mediante l'atteso A570, e quella di aggiungere schede acceleratrici interne costituiscano, anche per l'utente entry-level, dei validi motivi per preferire il 500 Plus.

Amiga 600 e Amiga 600HD

68000 a 7MHz
Chip custom: Agnus da 2 Mb, Super Denise, Gayle, Paula
Tecnologia SMD
1 Mb di Chip RAM (espandibile fino a 2 Mb di Chip e 8 di Fast)
1 Floppy interno da 3.5" 880K
Controller Hard Disk IDE su scheda madre
Hard disk interno da 2.5" da 20 (23 ms) a 120 Mb (di serie solo su 600HD)
Interfaccia per Memory Card (da 512K a 4 Mb)
Porta parallela, seriale, mouse, joystick
Porta interna per espansioni di memoria
Uscita RGB, videocomposita a colori, modulata
Kickstart 37.300 e Workbench 2.05
Accessori: alimentatore, mouse, 2 manuali (tre con l'A600HD), cavo TV
Dimensioni: 36x24x7 cm
Prezzi:
Amiga 600: Lire 669.000 + IVA
Amiga 600HD: Lire 800.000 + IVA

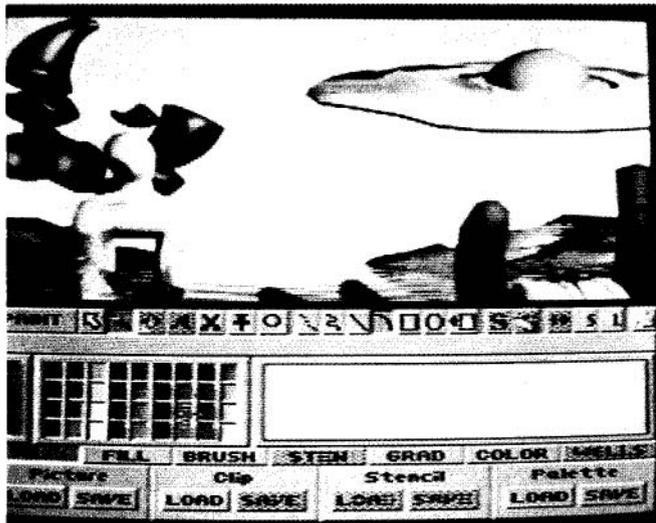
DCTV PAL

a cura della redazione

16 milioni di colori in videocomposito.

E' finalmente disponibile in Italia il DCTV in versione PAL, della statunitense Digital Creations, una scheda grafica esterna (adatta, quindi, a tutti i modelli Amiga) che è in grado di visualizzare immagini a 24 bit, cioè a 16 milioni di colori.
Prima di iniziare la nostra prova, sarà meglio cercare di capire cosa significhi e a cosa possa servire una scheda di questo tipo.
Occorre chiarire che il DCTV si distingue nettamente da schede come la Colorburst, l'HAM-E, l'Impact Vision, la Firecracker e tante altre: tutte queste schede hanno un'uscita RGB, cioè producono un segnale che è analogo a quello normalmente prodotto da Amiga e destinato a monitor RGB, come il 1084 della Commodore.
Il segnale RGB è di per sé estremamente nitido, pulito e quando non è interlacciato, solido come la roccia.
L'uscita del DCTV è, invece, videocomposita (come quella del Toaster).
E' il segnale utilizzato normalmente nel mondo video ed è per definizione interlacciato (sebbene possa anche non esserlo nel DCTV, ma con una certa perdita in qualità dell'immagine).
Il segnale videocomposito è meno nitido, non permette certi accostamenti di colore senza che comincino a "vibrare" o a produrre altri strani effetti (come accade di vedere qualche volta in TV).
Il videocomposito ama le transizioni di colore sfumate, i toni pastello, le mezze tinte.
E' molto adatto alle immagini fotografiche, quasi perfetto per quelle in movimento, si può avvicinare a certi metodi pittorici; è invece meno adatto alle immagini fisse e disegnate con contorni netti e contrasti vivaci.
Ciò non significa che il DCTV non possa visualizzare disegni, ma che bisogna porre una particolare attenzione nella scelta dei colori e degli accostamenti.
Fra le immagini dimostrative che accompagnano il DCTV c'è un bel disegno che dimostra come sia possibile raggiungere risultati ragguardevoli anche in modo videocomposito; c'è anche un'immagine 3-D che attesta la stessa cosa per i disegni in ray tracing; ma il massimo si raggiunge, senza dubbio, nelle immagini fotografiche o in quelle che si avvicinano maggiormente alle tecniche pittoriche classiche.

Ciò che rende il DCTV unico, è la possibilità di realizzare animazioni: nessuna scheda RGB per Amiga consente animazioni a 24 bit. Il DCTV sì; il motivo è presto detto: le immagini DCTV possono essere conservate in un formato proprietario che praticamente corrisponde al modo IFF a 4 bitplane (o a 3).
Sempre sui dischetti che accompagnano il DCTV c'è un'animazione da 60 frame (1/3 dello schermo) che occupa solo 450K.
Il risultato è eccezionale, se si pensa che un'immagine a 24 bit RGB può arrivare da sola a 1 Mb. 450 diviso 60 corrisponde a 7.5K per frame che, calcolando il fatto che l'animazione occupa 1/3 dello schermo, conduce a 22K per frame nel caso di un'animazione a tutto schermo. Proprio questa caratteristica ha fatto sì che il DCTV diventasse lo standard adottato dalla Commodore per il CDTV, come forse già sapete.
La versione per Amiga comprende, comunque, anche una sezione digitalizzatrice che non compare sulla scheda per il CDTV.
Il DCTV può dunque risultare particolarmente utile a chi produce animazioni, in primo luogo per visualizzarle, ma anche per registrare su nastro il prodotto finale. Grazie al DCTV si può dun-

Il programma DCTVPaint.

que evitare l'acquisto di un registratore a passo uno (non certo economico), a meno che non si abbia bisogno di un segnale in uscita di livello broadcast.

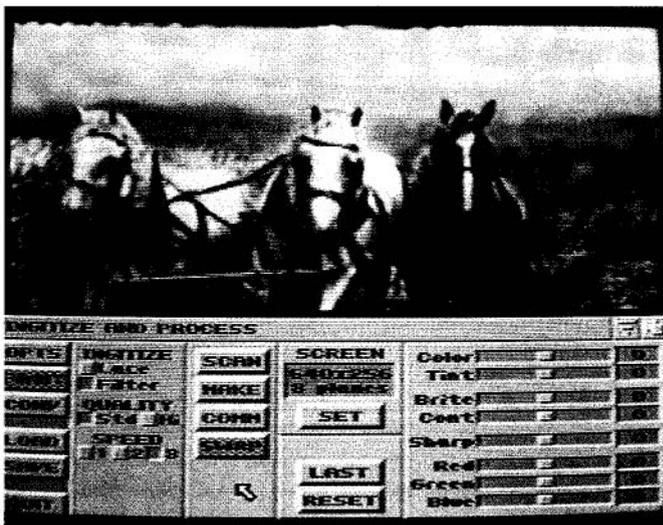
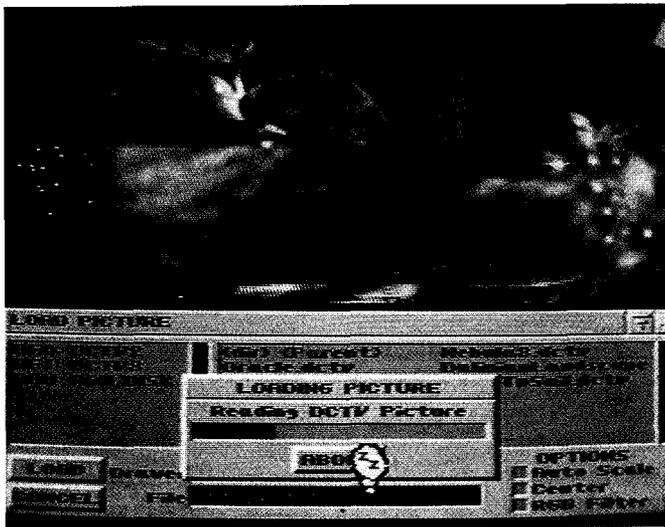
CONFEZIONE, ASPETTO ESTERNO E INSTALLAZIONE

L'allegra confezione del DCTV comprende quattro dischetti con la versione 1.1b del software, il DCTV, un manuale a spirale, un pieghevole di quattro pagine con le novità e l'errata corrige, una cartolina di registrazione.
L'apparecchio, molto piccolo, misura 13.5x9.5x3 cm e trova posto tranquillamente su qualsiasi scrivania.
Il suo aspetto è gradevole e accattivante.
L'installazione è semplicissima: spento Amiga, si connette il cavo dotato di connettore passante alla porta RGB di Amiga, il cavo del monitor RGB si collega al connettore rimasto libero. Si allaccia poi il secondo cavo del DCTV alla porta parallela (serve solo per digitalizzare le immagini).
Sul retro del DCTV appare un'uscita RCA (Video Out) cui andrà collegato il cavo (non fornito) per il monitor videocomposito o la TV dotata di presa Scart.
L'altra presa RCA del DCTV (Video In) è riservata a una telecamera o a un videoregistratore e serve per digitalizzare.
Il monitor può essere il 1084 o altri analoghi (come il Philips 8833), basta solo che siano dotati di tasto per passare dal modo RGB a quello CVBS.
In questo caso si potrà passare dal modo DCTV a quello standard mediante la semplice pressione del tasto citato.
Faccio notare che tutti i normali schermi Amiga (come quello del Workbench) restano visibili anche sul monitor videocomposito (ovviamente con il numero di colori standard) e possono anche sovrapporsi parzialmente ad esso.
Sul retro del DCTV compare anche un potenziometro (Pixel Adjust) che serve a tararlo.
Sul primo disco c'è un programma apposito, che comunque non abbiamo potuto usare perché, per un errore, il dischetto era in versione NTSC e non PAL (non preoccupatevi, negli esemplari in commercio dovrebbe comparire la versione giusta).
Comunque, non sembra affatto difficile tarare a mano il DCTV.
E' tutto: il DCTV si alimenta direttamente alla porta RGB di Amiga (non è quindi dotato di alimentatore separato) e, fra l'altro, risulta anche un dispositivo a basso assorbimento, visto che non scalda granché, pur essendo privo di ventola interna.
Una volta installato, lo si dimentica completamente, a meno che non si cominci a utilizzarlo.
Il DCTV non può essere usato di per sé con un Genlock, ma è possibile acquistare separatamente un convertitore DCTV-RGB, il cui segnale va inviato all'ingresso di un Genlock.
Non conosciamo la sua disponibilità in Italia, né se esista già la versione PAL.
In teoria, l'output videocomposito del DCTV può essere usato come input per un Genlock collegato ad un secondo Amiga: si potrà così miscelare, per particolari effetti (come il PIP), la grafica RGB e quella videocomposita.
E'anche possibile registrare su nastro l'output del DCTV e poi usarlo come input videocomposito per il Genlock, ammesso che si disponga di un secondo registratore.
Per installare il software si può tranquillamente usare l'Installer fornito (è il nuovo programma standard prodotto dalla Commodore per installare il software su hard disk o floppy).

I PROGRAMMI

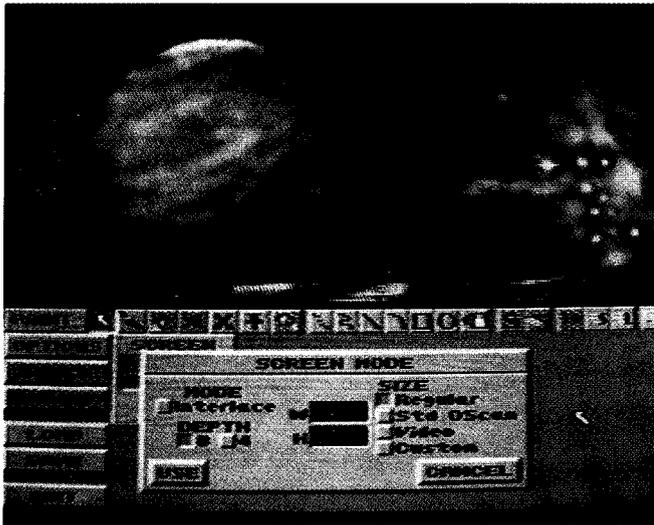
Nei quattro dischetti del DCTV sono compresi: il programma DCTV nella versione 1.1, che contiene la sezione di grafica pittorica, quella di digitalizzazione e quella di conversione fra formati.
Perché questo programma possa funzionare completamente in qualsiasi risoluzione occorrono 5 Mb di memoria. 3 Mb comunque bastano per far funzionare il DCTV utilizzando la versio-

Il programma DCTVProc.

Il caricamento molto veloce di una immagine.

ne, non ridotta, ma divisa in due parti, del programma, che si trova sul secondo disco.
Una parte (DCTVPaint) comprende il programmadi grafica pittorica, l'altra (DCTVProc),il digitalizzatore e la sezione di conversione dei file.
E' possibile far funzionare il programma anche con meno memoria (il minimo è 1 Mb), tuttavia certe funzioni verranno disabilitate per risparmiare memoria. Stando alle nostre prove, con 2.3 Mb (di cui 512K di Chip) e la versione ridotta del programma di Paint, erano inaccessibili le funzione Flood (ma non Fill), Undo, la gestione dei caratteri, la seconda Palette, quella degli Stencil e la Spare Page.
Tutti i programmi sono compatibili con 1.2, 1.3, 2.0 e seguono abbastanza fedelmente le regole di stile del 2.0.
Nei dischi si trovano anche altre piccole utility, programmi di pubblicodominio e immagini esemplificative.

LAVISUALIZZAZIONE

L'hardware del DCTV funziona sostanzialmente in due modi: come visualizzatore d'immagini a 24 bit (16 milioni di colori) e come digitalizzatore di tipo lento.
Sebbene il DCTV operi su file a 24 bit, i limiti del segnale videocomposito implicano al massimo 4 milioni di colori apparenti (21 bit).
Di fatto, sebbene la resa grafica anche di una scheda RGB a basso costo come la Colorburst sia indubbiamente superiore, i risultati ottenibili con il DCTV sono a dir poco soddisfacenti (specie in certe immagini). Se però i colori non rispettano i vincoli del modo videocomposito, si avràun immediato decadimento dell'immagine, dovuto principalmente a "vibrazioni" dei colori.
Pervisualizzare un'immagine IFF a 24 bit si puòoperare in più modi: il più diretto è quello di caricarla con uno dei programmi citati.
Si noti che le icone delle immagini fornite come esempio lanciano il programma del primo dischetto.
Se si è a corto di memoria, si possono caricare le immagini usando il requester del programma Paint o Proc del secondo dischetto, oppure cambiare le icone mediante Info del Workbench.
Il secondo modo richiede la conversione dell'immagine nel formato proprietario del DCTV.
Questo formato ha delle caratteristiche veramente speciali: prima di tutto è estremamente compatto. Un esempio per tutti: ho provato a convertire un'immagine IFF a 24 bit da 736x480 da 819910 byte nel formato .dctv: il risultato è stato un file da 147K. Le immagini caricate sul DCTV nei due formati appaiono del tutto identiche: non c'è dunque alcuna perdita di informazione per l'hardware videocomposito del DCTV.
Un'altra caratteristica del formato è che appare come un normale file IFF a 4 (o 3) bitplane.
Ciò significa che si può visualizzarlo utilizzando un qualsiasi programma (anche di pubblico dominio), capace di caricare un file IFF Hi-Res a 16 colori: l'immagine apparirà sul normale schermo RGB di Amiga come un confuso labirinto di pixel, ma se il DCTV è collegato alla porta video, questo intercetterà i dati in uscita, li decomprimerà in tempo reale e li convertirà in uno schermo videocomposito a 4 milioni di colori. Ho provato con Mostra, dell'italiano Sebastiano Vigna, e ha funzionato senza particolari problemi (lostesso può dirsi per le animazioni caricate con Rtap).
In certi casi è però necessario agire sui cursori per centrare l'immagine, prima di raggiungere il risultato sperato.
In teoria, è possibile usare i file .dctv con qualsiasi programma di Paint, lavorando sull'output del monitor videocomposito; l'unica condizione è che l'angolo in alto a sinistra dell'immagine non venga alterato (e che l'origine dell'immagine non venga spostata).
Il DCTV vi ripone, infatti, delle informazioni che servono all'hardware per riconoscere l'immagine come propria.
Si noti che la barra dei menu standard si sovrappone a queste informazioni, cancellandole e impedendo all'hardware tale riconoscimento.
E' anche possibile riconvertire un'immagine .dctv nel formato IFF24, ma si avrà una perdita di informazioni (ininfluenti in modo videocomposito, ma significative a livello RGB).
In entrambi i modi indicati, si otterrà una visualizzazione di buona qualità, limitata principalmente dalle caratteristiche intrinseche del segnale videocomposito e da quelle del dispositivo di output (non tutti i monitor e le TV sono uguali).
Con il DCTV si possono visualizzare anche animazioni in formato .dctv: per farlo, basta usare programmi di pubblico dominio, come View, fornito con il pacchetto.
E' dunque possibile creare animazioni a 24 bit mediante un programma di ray tracing, convertirle nel formato .dctv, creare un file in formato Anim (mediante, per esempio, il programma PD MakeAnim, sempre fornito su disco) e visualizzarle in 4 milioni di colori.
Ribadisco l'importanza di questa caratteristica (permette anche di registrare il risultato su qualsiasi videoregistratore), che non si tro-

Il pannello Screen Mode.

va se non su sistemi con quantità enormi di memoria, hard disk giganteschi, coprocessori grafici terribilmente veloci.

LA DIGITALIZZAZIONE

Per digitalizzare basta connettere una telecamera a colori (o un videoregistratore) alla presa Video In del CDTV, caricare la sezione Digitize, impostare alcuni parametri, controllare l'inquadratura sul monitor videocomposito e premere il pulsante sinistro del mouse. In meno di dieci secondi, durante i quali l'immagine deve rimanere fissa, si otterrà una digitalizzazione a 24 bit pronta per essere salvata.
E' difficile valutare la qualità della digitalizzazione: essa dipende in misura notevole da quella del segnale in ingresso. Le telecamere a colori (a differenza di quelle in bianco e nero) hanno normalmente una risoluzione piuttosto bassa che incide sull'esito finale. I milioni di colori disponibili rendono, comunque, possibili digitalizzazioni di qualità estremamente elevata, a giudicare dalle immagini fornite su disco. Certi risultati, però, richiedono segnali in ingresso di qualità professionale.

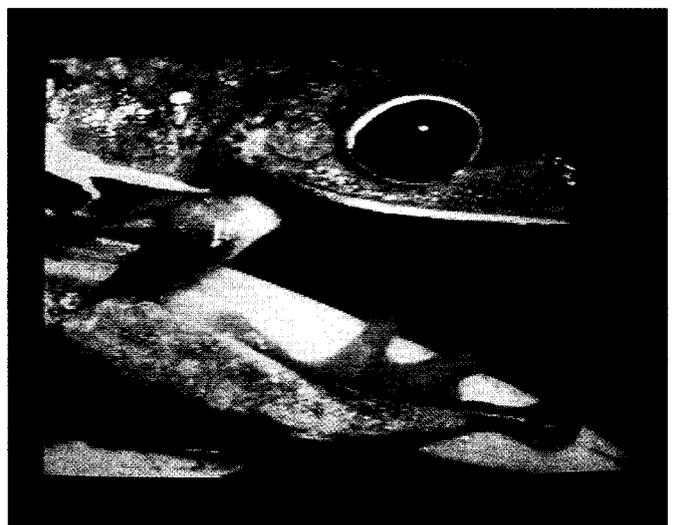
L'ELABORAZIONE DELLE IMMAGINI

Una volta catturata l'immagine, è possibile modificarla usando una decina di parametri controllabili mediante cursori: Color, Tint, Brite, Contrast, Sharp, Red, Green, Blue. In particolare, Tint permette di eliminare con facilità falsi colori dovuti a una luce troppo calda o troppo fredda. Sharp permette, invece, di aumentare la nitidezza dell'immagine e rendere più netti i contorni.

IL FORMATO DEI FILE E LA CONVERSIONE

Le immagini possono essere salvate (e caricate) direttamente in tre formati: raw (dati grezzi), .dctv e IFF 24. Il formato raw è praticamente una versione compressa del formato a 24 bit usato internamente dal DCTV: è consigliabile usarlo per salvare le immagini su cui si sta lavorando.
La sezione Convert del programma permette di trasformare le immagini a 24 bit in diversi formati Amiga (e viceversa) per utilizzarle con il normale software Amiga. I formati supportati sono tutti quelli standard: Lo-Res, Hi-Res, HAM, Half Brite. L'overscan arriva a un massimo di 736x566 pixel (PAL). Non sono implementati i modi grafici dinamici e l'HAM in Hi-Res.
E' anche possibile usare il dithering per aumentare il numero di colori apparenti. Sono supportati tre modi: Ordered, Diffuse e Average. Si può operare direttamente sulla palette dei colori usata per l'immagine di arrivo, scambiando colori fra loro, modificando i valori RGB, e bloccando qualche colore prima dell'elaborazione. E' infine possibile eliminare il colore 0 per evitare l'effetto trasparenza quando l'immagine venga gestita da un genlock.
Esistono anche programmi CLI per la conversione dei file, utili per creare file batch, magari in ARexx, che operino automaticamente: sono IFFTODCTV e S4DTOILBM (per passare dai tre file RGB generati da Sculpt4D al formato IFF24).

VERSO LA PITTURA

Disegnare in videocomposito non è come disegnare in RGB. Dipingere con DCTVPaint non è dunque come usare Deluxe Paint. Non si ha a che fare con pixel, ma con "onde di colore", non con componenti RGB, ma con crominanza e luminanza. Di fatto, si possono ottenere quasi gli stessi risultati ottenibili in RGB (a 24 bit), ma il metodo da usare è profondamente diverso. Il disegno in videocomposito si avvicina di più a forme tradizionali di pittura, come tempera, acquarello, olio e il Paint del DCTV possiede funzioni che tendono a simulare il più fedelmente possibile tali metodi di pittura. Il prodotto, dunque, risulta destinato non tanto a chi vuole creare qualche cerchio o linea colorata, ma a chi intende produrre oggetti artistici, è dotato di un'autentica predisposizione alla grafica pittorica e vuole cimentarsi con un nuovo tipo di supporto: il videocomposito. Questo

non significa che il programma non possa essere usato per obiettivi più limitati: anche Superbase 4 può essere destinato a gestire l'elenco di una dozzina di numeri di telefono.
Sia chiaro, tuttavia, che molte delle regole e delle abitudini acquisite con programmi RGB non hanno più valore in un ambiente videocomposito, mentre esistono altre limitazioni (certi colori tendono a scomparire altri a vibrare, alcune combinazioni generano con facilità sfarfallio o righe inaspettate...) con cui bisogna imparare a convivere. D'altraparte, per fare solo due esempi, il fastidioso effetto di scalinatura, tipico del mondo RGB qui è molto meno grave e le transizioni di colore rasentano la perfezione.
Il programma destinato al disegno appare completo, funzionale e veloce, anche con un 68000 (si opera sempre con milioni di colori). Mi è sembrato di gran lunga il programma di grafica a 24 bit più veloce che abbia mai provato. Ci preme sottolineare che il prodotto ha già raggiunto una notevole maturità sotto tutti i punti vista: è praticamente unanime il coro di coloro che sostengono che il Paint del DCTV sia il miglior programma di grafica pittorica disponibile per Amiga, superiore allo stesso Deluxe Paint. Il confrontocon l'unico programma analogo esistente per Amiga, il Paint a 24 bit del VideoToaster (che produce output videocomposito come il DCTV), gioca tutto afavore del DCTV: al di là di tutto, il programma del Toaster, infatti, opera su uno schermo HAM, qui si disegna direttamente in video composito con milioni di colori.
Inoltre, alcune delle soluzioni adottate per la primavolta da questo programma, come la tavolozza per la miscelazione dei colori, sono state imitate da recenti versioni di altri pacchetti di grafica pittorica (uno fra questi è proprio Deluxe Paint IV). Attualmente, il Paint del DCTV rappresenta lo stato dell'arte nella grafica a 24 bit.
Al Paint del DCTV manca tuttaviaqualsiasi funzione di animazione: per crearle occorre generare i frame uno per uno e poi usare un programma esterno per costruire l'animazione.
Se la prima versione del programma ha suscitato tanti entusiasmi (si veda l'analisi condotta alle pagine 21 eseguenti del numero25 di Amiga Magazine), la seconda, cioè la 1.1, non può che confermarli: sono state aggiunte, infatti, proprio le funzioni di cui più si era sentita la mancanza in precedenza, come l'Undo (che richiede 1 Mb in più di memoria), il supporto per font colorati (anche sotto 1.3) e font outline del 2.0, una gestione più facile dei testi, la capacità di importare brush di Deluxe Paint.
Non è possibile elencare le centinaia di funzioni disponibili, anche perché le funzioni si possono combinare tra loro in quasi tutti i modi, e il numero di combinazione possibili diventa praticamente sterminato. Nonostante la grande complessità, il programma appare decisamente user-friendly, semplice da usare, almeno ai primi livelli, emolto intuitivo.
Ci limiteremo, dunque, a un semplice elenco che possa dare una (pallida) idea del programma.
In primo luogo, segnaliamo che se la memoria lo permette, si possono avere due pagine in memoria: quella secondaria è detta Spare Page e può essere usata per disegnare sulla pagina principale (le opzioni chiamate RubThru fanno riferimento ad essa). Inoltre, è presente l'opzione Undo che annulla l'ultima operazione effettuata (è quasi indispensabile, perché molte delle funzioni del programma, vista la particolare complessità di certe operazioni, generano effetti non sempre pienamente prevedibili).
I modi grafici fondamentali sono tre: Solid (opera sia sulla luminanza che sulla crominanza), Tint (opera solo sulla crominanza) e Shade (opera solo sulla luminanza). Ci sono, inoltre, sei stili: Airbrush, Blend, Rubtrhu, Watercolor, Shade, Smooth, Filter. L'ultimo serve solamente a ridurre le combinazioni di colore illegali in videocomposito.
I pennelli hanno otto forme base la cui ampiezza è definibile mediante un cursore. Di tutti i pennelli si può definire il rapporto tra altezza e larghezza (il brush quadrato si può trasformare così in un rettangolo). Il parametro numerico Flow determina, essenzialmente, la "quantità" di colore che fuoriesce dal pennello, più è alto e maggiore sarà la capacità della "pennellata" di coprire il colore sottostante (il comportamento varia anche in funzione del modo grafico selezionato e dello stile).
I pennelli, in questo caso detti Clip, possono anche essere ritagliati dallo schermo (in molti modi diversi) o importati come file .clip. Una volta definita, la clip può essere ruotata di qualsiasi grado e deformata secondo l'asse delle X o quello delle Y mediante tre funzioni: Flip, Bend, Shear. La clip può essere ridimensionata in tutti i modi.
Si può anche aggiungergli un'ombra, definendone distanza, angolo, opacità.
Il pannello Fill serve a definire il modo in cui devono essere riempite le aree; ci sono quattro modi principali: Solid, Gradient, Pattern, Warp.
Ognuno di essi ha delle subopzioni per un totale di 18 modi diversi.
Gli effetti ottenibili sono veramente potenti e sono in grado di operare con la clip

corrente: segnalo solo il modo Gradient-Remap che permette di trasformare qualsiasi gradiente di luminosità in un gradiente di colori completamente definibile dall'utente, e i diversi modi Warp che generano effetti prospettici, avvolgendo la clip su cilindri e sfere la cui curvatura è definibile dall'utente.
L'ultimo modo è l'Highlight: permette di usare la clip come un filtro colorato per una luce ideale che viene proiettata sullo schermo.
Il pannello Gradient consente di definire il gradiente da utilizzare per i Fill: si opera inserendo dei colori come se fossero dei TAB su una linea.
I gradienti (come le clip, le palette e gli Stencil) possono poi essere salvati su disco. Per definire i colori esistono cinque modi diversi: il più tipico è quello della tavolozza in cui i colori possono essere mescolati tra loro per produrne altri. Gli altri quattro usano cursori per definire valori RGB, CMY, HSV, HLS. Lo Stencil, una maschera da usare per le operazioni di disegno, può essere creato aggiungendo o sottraendo ciò che si disegna sullo schermo, oppure definendo, mediante valori di crominanza e luminanza, una gamma di colori cui verrà applicato. Lo Stencil può essere reso opaco, nascosto o translucido.
Il pannello Text permette di scegliere il font (anche outline sotto 2.0) e di scrivere il testo entro un gadget stringa.
I font colore possono essere usati sotto 1.3 mediante il programma esterno ColorText (si trova in Deluxe Paint o nei dischi di font). Una volta creato, il testo può essere usato come qualsiasi altra clip (o come Stencil). Si può ovviamente scegliere lo stile e anche quattro differenti aspect ratio.
Lo Zoom (2x,4x o 8x) è molto potente: tutte le operazioni restano abilitate in modo Zoom; inoltre, in videocomposito, gli ingrandimenti non generano solo cubetti irriconoscibili (come avviene in RGB), per cui risultano molto più utili.
In conclusione, il programma ha delle funzioni veramente potenti e, una volta superato lo sconcerto iniziale generato dal non aver più a che fare con i "mosaici di pixel" tipici dell'RGB, difficilmente si vorrà far ritorno ai soliti modi grafici di Amiga.

LA DOCUMENTAZIONE

Elegante, sintetica, ma precisa e priva di lacune. Con semplici tutorial per ogni sezione del software, indice degli argomenti, indice dei tasti, indice delle icone (!) e appendici di approfondimento. Il manuale, rilegato a spirale, è stato completamente aggiornato alla versione 1.1. Il difetto maggiore è che è in inglese.

CONCLUSIONI

Praticamente, gli unici effettivi limiti del DCTV sono: la qualità del segnale videocomposito (d'altra parte, il pubblico cui si rivolge non è quello che ha bisogno di un'uscita a livello broadcast), la mancanza di un uscita S-VHS e quella del supporto ARexx. Il prodotto è curato in tutti i particolari; il programma in grafica pittorica non ha praticamente rivali e appare realmente affascinante (molti utenti del Toaster, in USA, comprano il DCTV solo per usare il suo Paint); la possibilità di creare animazioni (sebbene non supportata direttamente dal Paint) lo rende insostituibile in certe situazioni; è capace anche di digitalizzare: che dire di più?
Qualcosa ancora si può: il numero di programmi che lo supportano è già consistente (se non altro come qualità) ed è destinato sicuramente ad aumentare: Real 3D, Imagine, Draw4D-Pro e Caligari 2.0; Scenery Animator e VistaPro; Art Department Professional e Image Master. ▲

SCHEDA PRODOTTO

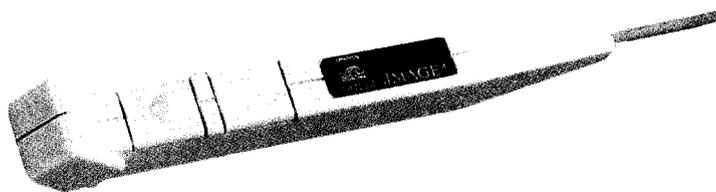
Nome Prodotto: DCTV
Casa Prodruttrice: Digital Creations
Venduto da: AP&S - Corno di Rosazzo (UD) Tel. 0432/759264
Prezzo: Lire 1.190.000
Giudizio: eccellente
Configurazione richiesta: qualsiasi Amiga, minimo 1 Mb, meglio 5 Mb
Pro: programma di grafica pittorica, formato compreso dei file, animazioni, velocità del software;
Contro: assenza segnale S-VHS, macro e ARexx
Configurazione della prova: A500 1.2, 1.3, 512K Chip, 1.8 Mb Slow-Fast

"THE BRUSH"

Gabriele Stecchi

Un'insolita alternativa al mouse.

Nell'affollato e, a volte, confuso mondo dell'informatica di consumo, èfacile imbattersi in curiose periferiche, semplici optional o sofisticati potenziamenti che siano. Qualsiasi mercato, ma soprattutto quello americano. è fornitissimo di tali prodigi della tecnica, pronti a dare una mano sia all'appassionato che al serio professionista. "The Brush", letteralmente "il pennello", appartiene proprio a questa categoria di originali aggeggi. In parole povere si tratta di un metodo di interazione con l'interfacciagrafica di Amiga sostitutivo del mouse, basato sullo stesso principioe collegabile nella normale porta 1. Vedremo le sue potenzialità nel dettaglio e capiremo se il suo acquisto potrà cambiare, e come, il modo di disegnare e comunicare con la macchina

LA CONFEZIONE

La robusta confezione, già dal primo sguardo, non lascia dubbi sulla natura del contenuto; campeggia, infatti, sulla sua facciata l'immagine di una mano che impugna una sorta di grosso pennarello dotato di due pulsanti e con un filo ad un'estremità.Sul retro della confezione una lunga lista di specifiche e alcune dettagliate tabelle ci illustrano la meccanica e il principio di funzionamento di "The Brush"; niente di trascendentale, comunque, poiché questo pennello è un'interfaccia optomeccanica che, espressioni roboanti a parte, funziona esattamente come un mouse. Sempre sulla confezione troviamo un elenco di ciò che contiene il pacchetto, vale a dire una copia di DPaint II con manuale, una garanzia, un "porta pennello" e naturalmente "The Brush". Per quanto riguarda il software incluso, è composto da due dischi, rispettivamente contenenti la seconda release del popolare programmae il relativo Art Disk, mentre siamo rimasti abbastanza sorpresi nello scoprire che il "porta pennello" altro non è che un curioso e massiccio pezzo di legno laccato e forato. Installare "The Brush" non ha comportato alcuna difficoltà, dato che l'operazione consiste nel sostituire al mouse in dotazione il pennello inserendone il connettore nella porta 1. Il suo utilizzo è altrettanto elementare dato che è gestito dal sistema esattamente come la sua controparte tradizionale. "The Brush" non nasce però come sostitutivo integrale del mouse, dato che la sua impostazione prettamente grafica, finalizzata più alla precisione che alla comodità, lo rende un po' goffo in ambienti di lavoro diversi dai programmi di disegno. Le prestazioni della periferica, testata con programmi quali DPaint II e successive versioni, sono paragonabili a quelle del mouse standard di Amiga. Se da una parte l'impugnatura può risultare più congeniale ad una parte degli utenti, da un'altra lo strumento può risultare decisamente scomodo nell'interazione con l'interfaccia grafica di sistema, nella fattispecie con i menu adiscesae con i gadget, proprio in virtù della sua velleità di maggior precisione che costringe l'utente stesso a numerosi spostamenti del puntatore per il loro raggiungimento. Preso atto di tali pro e contro "The Brush" appare come uno strumento di disegno adatto ad un uso educativo con programmi dedicati all'infanzia, dove potrebbe essere preferito al meno intuitivo mouse, o come un'originale alternativaal mouse in combinazione con software appropriato.

SCHEDA PRODOTTO

Nome Prodotto: The Brush
Casa Prodruttrice: Golden Image
Distribuito da: Armonia Computers - Tel. 0438/435010
Prezzo: non comunicato
Giudizio: mediocre
Configurazione richiesta: non necessita di alcuna configurazione particolare
Pro: improntato ad una maggiore precisione
Contro: poiché studiato per applicativi grafici può risultare scomodo se utilizzato come sostitutivo integrale del mouse
Configurazione della prova: Amiga 500

I LASERGAME

Maurizio Miccoli & Vittorio Calzolari

Finalmente è possibile portarsi a casa, ad un prezzo accessibile, i grandi classici che hanno fatto la storia dei videogiochi da bar al laser!

Chissà quanti di voi hanno fatto la fila per poter giocare in sala giochi alla mitica meraviglia di Don Bluth, Dragon's Lair, e ai suoi purtroppo pochi successori?!
E che dire dei più giovani, che ne hanno sentito solo parlare, come una leggenda, da coloro che possono dire "c'ero anch'io"?
Bene, oraqualcuno, lassù in Germania, ha avuto l'ennesima "ideafantastica": utilizzare un comune lettore laser per leggere i laser disk game, elaborare il software necessario per far gestire il tutto da un Commodore Amiga 500 e, infine, assicursi i diritti dei più noti laser game.
Il sistema fortunatamente è stato prontamente importato in Italia dalla Newel (famoso negozio di Milano), che attualmente ne è anche l'unico distributore.

IL "PACK" PER OGNI ESIGENZA

La Newel ha commercializzato una combinazione (POWER PACK) che comprende un multilettore Pioneer CLD 1600, un disco di software, l'interfaccia e il disco di Dragon's Lair a 1.390.000 lire; se si possiede già il lettore è possibile acquistare solamente il resto (BASIS PACK) a 390.000lire.
E' disponibile anche un'interfaccia a infrarossi (molto comoda) al prezzo di 250.000 lire.

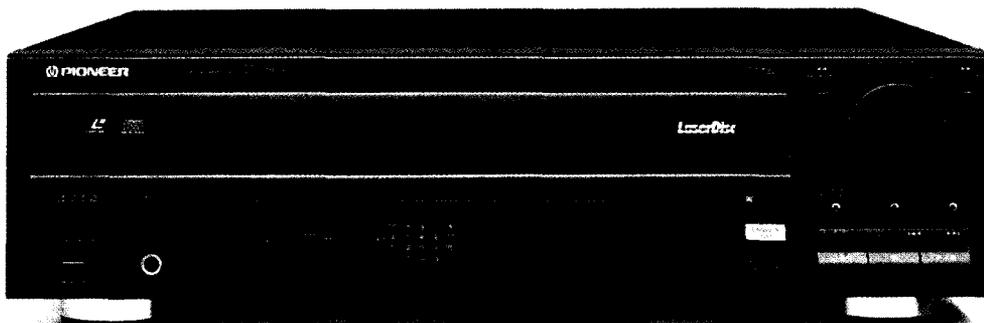
COME PARTIRE...

La sequenza di installazione è la seguente: a computer spento, si inserisce il connettore a 25 poli dell'interfaccia nella porta parallela di Amiga, connettendo poi l'altro capo alla presa "control-in" del lettore laser.
E' necessario disporre di un monitor o televisore dotato di presa Scart e ingresso video composito, oppure di due prese scart. Nel caso di utilizzo di un monitor 1084 o 8833, vi consigliamo di effettuare il collegamento come segue:

1) Uscita Scart del lettore laser all'ingresso Scart del monitor;
2) Uscita video monocromatica di Amiga all'ingresso marcato "C" del monitor.

Questo è necessario perché, probabilmente, le masse del circuito digitale di controllo e quello analogico video del lettore sono separate. Inserite il dischettocontenente il programma di controllo e accendete il computer.
Unavoltaeseguitala procedura di bootstrap, verranno visualizzate diverse schermate di istruzioni che vi guideranno nel gioco.
Il programma di controllo gira soltanto con laversione 1.3 del sistema operativo, per cui i possessori di Amiga 3000 o 500 Plus dovranno procurarsi tale versione.
Se volete divertirvi a "scorrere" il gioco, per vedere cosa succede senza dover giocare, è sufficiente togliere il jackdal "control in", abilitando così il telecomando.

MANO AL JOYSTICK...

Dragon's Lair narra la vicenda della splendida principessa Daphne, rapita dal cattivissimo Dragone, Signore di un castello incantato.

Il lettore CLD-1600.

Il retro del CLD-1600.

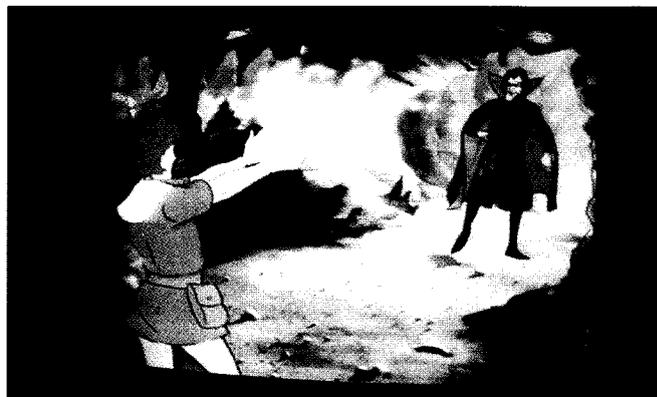
Dragon's Lair.

Naturalmente noi dobbiamo vestire i panni di Dirk the Daring, che deve attraversare ben 37 ambienti popolati dalle più fantastiche "avversità", per potersi avventurare nello Scontro Finale nella Tana del Dragone.
Qui troverà una spada magica, con cui potrà trafiggere il perfido e liberare dalla sfera magica che la tiene prigioniera la sua amata.

...O ALLA TASTIERA

Il secondo laser game disponibile a tutt'oggi è Thayer's Quest, un'adventure simile ad un libro game. Infatti, il gioco presenta delle sequenze interrotte da punti-chiave in cui bisogna scegliere, tramite un menu, che direzione prendere, che oggetto utilizzare o cos'altro fare.
Lo scopo del gioco è aiutare un discepolo dello Stregone Bianco Druce ad attraversare le cinque terre per trovare i pezzi che compongono un talismano in grado di sconfiggere il terribile mago Sorsabal, che domina col terrore quel mondo.

E PER IL FUTURO...

Per inizio aprile dovrebbe finalmente arrivare Space Ace, a cui faranno seguito Orpheus, Shadow of the New Stars e Voyage to the New World nell'immediato futuro.
Successivamente sono previsti anche Cobra Combat, Thayer's Quest 2, Casino Royal 1 e 2 (V.M. 18), MACH 3 e Firefox.

GIUDIZIO FINALE

Si tratta di un prodotto relativamente costoso, ma al riguardo basterebbe ricordare i prezzi di alcune cartucce per console, che non possono certo vantare la medesima qualità. Eliminati i problemi di collegamento e/o di interferenze sembra di essere in sala giochi, senza dover sborsare pacchi di gettoni! ▲

SCHEDA PRODOTTO

Nome Prodotto: Sistema LDG
Casa Prodruttrice: Software Corner
Distribuitore: Newel - Milano - Tel. 02/323492
Prezzo: Power Pack Amiga L. 1.390.000
Giudizio: buono
Configurazione richiesta: Amiga qualsiasi modello con porta parallela e kickstart 1.3
Pro: alta qualità della grafica e del sonoro.
Contro: leggera sabbiosità dell'immagine, possibili problemi di interferenze e di collegamenti, almeno inizialmente.
Configurazione della prova: Amiga 500 kickstart 1.3

VIDEO DIRECTOR

Gabriele Ponte

Una moviola nell'Amiga.

Il mondo delle periferiche dedicate ad Amiga è in continuo fermento e, dopo l'arrivo di schede video a 24 bit, digitalizzatori audio e video, Genlock, schede acceleratrici e hard disk, ecco apparire un prodotto che è in grado di trasformare Amiga in un'autentica stazione di montaggio video. La Gold Disk, già nota per la produzione di software ad alto livello (ricordiamo Professional Page, Professional Draw, Professional Calc, Hyperbook e Showmaker) ci presenta ora Video Director, un prodotto software-hardware che consente, a chiunque disponga di una telecamera e di un videoregistratore, di montare con relativa facilità filmati video anche molto complessi.

LA CONFEZIONE

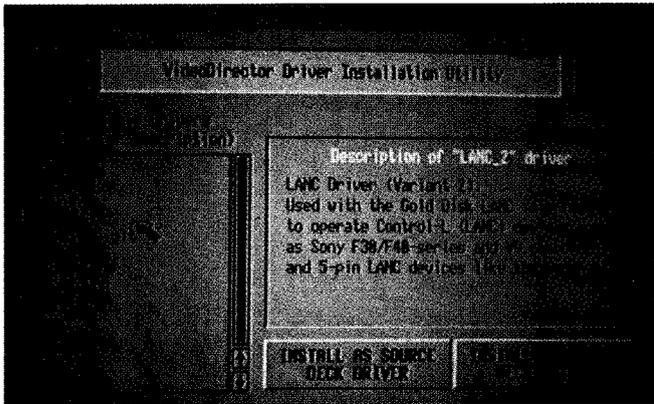
All'interno dell'elegante confezione troviamo due cavetti: il primo, da collegare alla porta seriale di Amiga, permette di pilotare una telecamera o un videoregistratore dotati di presa LANC o Remote; il secondo, che va collegato alla presa joystick, è un emettitore-digitalizzatore di segnali all'infrarosso, che permette di emulare le funzioni svolte dai normali telecomandi al fine di controllare un secondo videoregistratore. I due dischi presenti nella confezione contengono il software di gestione delle interfacce. Il manuale in inglese, rilegato con una spirale, è veramente esauriente.

LA CONFIGURAZIONE

La configurazione minima necessaria per utilizzare Video Director comprende: Amiga con 512K di memoria, un monitor o un televisore, una telecamera o un videoregistratore da utilizzare come sorgente video (possibilmente dotati di presa LANC o Remote), un videoregistratore su cui riversare il risultato finale.

Per un uso più avanzato del sistema, bisogna aggiungere: un Genlock (nel manuale si raccomanda l'uso del SuperGen 2000S), per effettuare anche il montaggio di titoli o di animazioni; un secondo monitor o televisore, per visionare l'immagine proveniente dalla sorgente video; una scheda multiseriale, da collegare ad Amiga nel caso si vogliano utilizzare due sorgenti video provviste di porta seriale.

La configurazione da noi utilizzata era così composta: Amiga 500 1.3 con 2 Mb di memoria e hard disk da 20 Mb; Camcorder Sony CCD F500E, usato come video sorgente e collegato tramite l'interfaccia seriale fornita con Video Director alla presa Remote della telecamera; videoregistratore Sony SLV-715, utilizzato come video destinazione; televisore a colori Philips

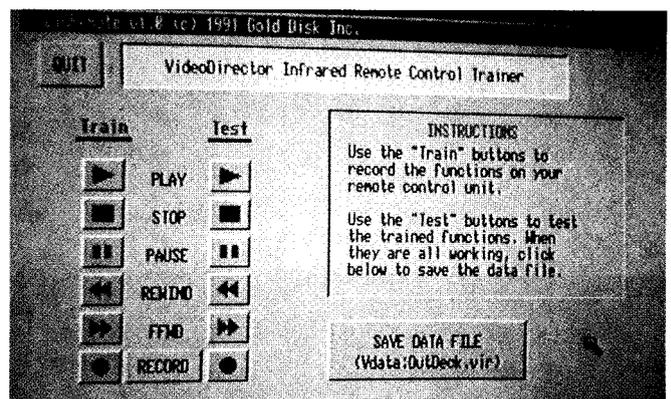
***Video Director** è un prodotto software-hardware che trasforma Amiga in una centralina per il montaggio video.*

dotato di presa Scart, con cui potevamo visionare sia lo schermo Amiga (collegato alla presa Scart), che il video della telecamera, commutando il televisore in modo antenna oppure Scart (il segnale proveniente dalla telecamera transitava attraverso il videoregistratore e arrivava alla presa antenna del televisore).

L'INSTALLAZIONE

L'installazione dell'hardware varia a seconda della dotazione dei componenti video: se la propria cinepresa, o il proprio videoregistratore, dispone di un ingresso LANC o Remote, potremo collegarvi il primo cavetto (un minijack stereo), che termina alla porta seriale di Amiga. Il secondo cavetto, da collegare alla porta joystick 2 di Amiga, permette, invece, di inviare segnali all'infrarosso per pilotare il secondo videoregistratore. Bisogna, inoltre, collegare la telecamera al videoregistratore con i normali cavetti audio e video.
Terminati questi collegamenti, da farsi sempre a computer spento, si possono alimentare e varie periferichee, sempre per ultimo, il computer.
Si può ora cominciare a lavorare con una copia del software: troveremo tre programmi nella directory Video-Director del dischetto principale.
VideoDirector serve per il montaggio vero e proprio del filmato e per creare il database video, mentre gli altri due, InstallDrivers e TrainRemote, sono delle utility per la corretta installazione del sistema.
InstallDrivers permette di installare il drive per il pilotare i componenti video collegati (foto 1).
Nella tabella si troverà un elenco delle telecamere e dei videoregistratori attualmente supportati da Video Director.
TrainRemote consente di digitalizzare i segnali all'infrarosso del videoregistratore (foto 2): ciò consentirà al programma principale di pilotare il componente in maniera automatica. Vediamo più da vicino le sue caratteristiche. Il pannello alla sinistra dello schermo contiene i tipici comandi di un videoregistratore (Play, Stop, Pause, Rewind, Forward e Rec). E' possibile configurarli a proprio piacimento; del tasto Record si può decidere, ad esempio,

Foto 1: l'utility per l'installazionedel driver.

in che modo debba ricominciare la registrazione dopo una pausa dovutaalla ricerca di uno spezzone nel video sorgente: le opzioni disponibili sono Record (in tal caso, viene inviato nuovamente il segnale di registrazione), Play o Pausa. E' dunque possibile adattare il software al componente video impiegato. Si può inoltre stabilire la durata del segnale (da 1 a 3 secondi), mentre un'ulteriore opzione, presente nei menu a tendina, permette di selezionare sino a cinque velocità per la trasmissione del segnale all'infrarosso.
I comandi a sinistra dello schermo permettono, invece, di campionare i segnali emessi dal telecomando del videoregistratore e di verificare che l'operazione di campionamento sia avvenuta correttamente. Per campionare il comando di Play, ad esempio, bisogna selezionare il gadget PLAY, porre il telecomando del video a qualche centimetro di distanza dal ricevitore-emettitore (quello collegato alla presajoystick) e premere il tasto Play del telecomando: un messaggiosullo schermo confermerà o meno la riuscita dell'operazione. A questo punto, possiamo rivolgere l'emettitore connesso al computer verso il videoregistratore e premere il gadget PLAY nella colonna di destra, verificando se il componente video si sia effettivamente posto in modo riproduzione. Questaoperazione va ripetuta per tutti i comandi presenti sullo schermo. Una volta terminate queste operazioni, il programma salverà i dati in un apposito file, che verrà letto automaticamente da Video Director (TrainRemote, come del resto anche InstallDriver,va quindi usato solamente la prima volta che si utilizza il pacchetto oppure quando si cambiano i componenti video).
Se riuscite a pilotare entrambi i componenti video con l'hardware fornito dalla Gold Disk, potrete utilizzare Video Director nel migliore dei modi, cioè in modo Automatico; se, invece, non riuscite con uno solo dei due, dovrete operare in modo Semi-Automatico (dovrete cioè attivare manualmente le funzioni che non riuscite ad attivare in modo automatico); se nessuna delle due sorgenti risponde alle caratteristiche richieste da Video Director, dovrete operare in modo Manuale.

Foto2: TrainRemote campiona i segnali dei telecomandi all'infrarosso.

VIDEO DIRECTOR

Se il sistema funziona in modo Automatico e si possiede anche un Genlock, si può affermare che le caratteristiche di Video Director sono in grado di soddisfare anche le esigenze di un montatore video semi-professionista.
Un sistema così configurato permette, infatti, il montaggio automatico di una cassetta video, l'inserimento automatico di titoli o di immagini provenienti da computer e la creazione di un database di cassette videoe di tagli (clip) utilizzabili nei vari montaggi.
Una volta lanciato il programma, se il sistema non riconosce le periferiche installate con InstallDriver,un requester ci chiederà se vogliamo proseguire; i casi sono due: o il componente video collegato tramite la porta seriale non è acceso, oppure la porta stessa potrebbe essere danneggiata.
Se invece il sistema è installato correttamente, verranno caricati i driver che controllano i due componenti videoe apparirà lo schermo principale di Video Director.
Come potete vedere nella foto3, lo schermo è diviso in tre zone distinte: la prima è riservata al nome della cassetta e allo spezzone di nastro corrente; la zona di destra contiene le funzioni di controllo dei componenti video collegati e può variare a seconda del driver installato; la terza (nella parte inferiore dello schermo) riguarda la struttura del filmato che verrà riversato sul video destinazione.
Prima di dare avvio alla fase di editing, è bene controllare il funzionamento delle varie funzioni di trascinamento del nastro (Play, Stop ecc.) mediante i comandi presenti nella seconda zona.
Supponiamo ora di voler effettuare un piccolo montaggio video, operando in Modo Automatico.
Dopo aver inserito nel componente video sorgente una cassetta con il filmato da montare e una cassetta vergine in quello destinazione, potremo selezionare il gadget TAPE???: un requester ci permetterà di inserire il nome del nastro nel database, oppure di richiamare i dati memorizzati in precedenza.
Conviene riavvolgere completamente il nastro e selezionare il gadget RESET per azzerare sia il contatore del video che quello del programma.
Possiamo quindi far scorrere il nastro in avanti per cercare la prima scena da memorizzare: si possono utilizzare sia i comandi della telecamera, che quelli sullo schermo di Video Director (Play, Forward, Rewind ...).
Una volta trovata la scena che cerchiamo, selezioniamo il gadget SET (START) per memorizzare il punto iniziale e quindi SET (END), unavoltatrovata la fine dello spezzone.
Dopo aver assegnato un nome alla scena, possiamo anche salvarla nel database (opzione CAVE), oppure salvarla e aggiungerla automaticamente alla lista che appare nella finestra inferiore, riservata all'assemblaggio del video finale (opzione ADD).
Ripetiamo questo procedimento per ogni taglio di montaggio del nostrovideo.
Così facendo, andremo creando, nella parte inferiore dello schermo, la lista di eventi che verranno poi assemblati nel video finale.
La lista può essere modificata sia cambiandone l'ordine, sia eliminando o aggiungendo nuovi clip, sia richiamando il singolo spezzone per modificare i valori che ne determinano il punto iniziale o quello finale. Possiamo anche aggiungere spezzoni provenienti da altre cassette, prelevandoli dal database presente su disco, se erano stati memorizzati in precedenza. Se la lista diventasse troppo lunga, possiamo riunire diversi spezzoni in una singola Scena (è possibile, comunque, scorrere la lista con gli appositi gadget, oppure usare il modo interlacciato per visualizzare un numero doppio di linee). Il programma memorizza anche il valore del contatore di una cassetta utilizzata; ciò per-

Foto 3: lo schermo principale di Video Director.

VIDEO SORGENTE

INTERFACCIA	PROTOCOLLO	PERIFERICA
Gold DiskLANC	LANC	Camcorder e video registratori dotati di presa Remote
SONY V-BOX	LANC	Camcorder e video registratori dotati di presa Remote
Cavo Seriale	NEC Serial	Nec PC-VCR
Cavo Seriale	Panasonic 5PIN	Selectta AG-1960 RS
VuPort Selectra	Panasonic 5PIN	Panasonic AG-1960

VIDEODESTINAZIONE

L'interfaccia all'infrarosso può controllare la magglor parte dei videoregistratori in commercio. Se il proprio sistema video non e supportato da nessuna di queste interfacce, si dovrà operare in MODO MANUALE.

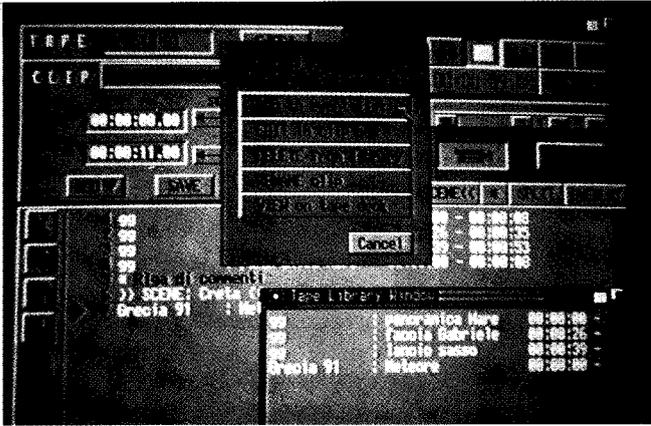



Foto 4: la creazione di un database video è certamente una delle funzioni più importanti del programma.

mette di recuperare il valore del contatore ed evita di dover riavvolgere la cassetta. In ogni caso, sono presenti degli appositi comandi per ricalibrare nastro e contatori.
Se si possiede un Genlock, si possono inserire nella lista degli eventi di Overlay: un apposito requester permette di indicare il nome dello schermo da richiamare, mentre due cursori consentono di impostare la durata della dissolvenza sia in ingresso (FADE IN) che in uscita (FADE OUT).
A questo punto, non resta che dare il comando Assem per riversare il tutto sul video destinazione: se i vari tagli fanno parte di un'unica cassetta, potremo anche andare a bere un caffè, mentre Video Director si pone alla ricerca dei vari spezzoni nel sorgente; se invece avessimo usato spezzoni che appartengono a cassette diverse, il programma si fermerà e ci comunicherà quale nastro dobbiamo inserire nel componente video sorgente, prima di riprendere il montaggio.
Tutto questo avviene se si riesce a utilizzare il sistema in modo Automatico.
Purtroppo, non siamo riusciti a pilotare mediante il dispositivo all'infrarosso il videoregistratore Sony SLV-715 HI-FI (mentre ha funzionato con un vecchio Philips a due testine), per cui abbiamo dovuto operare in modo Semi-Automatico: la telecamera era guidata dalla porta seriale tramite il Remote, per cui rispondeva a tutti i comandi sia per i tagli che per la memorizzazione del contatore, mentre per il videoregistratore abbiamo dovuto installare il driver Manuale.
In fase di riversamento del filmato, un conto alla rovescia ci avvisava quando dovevamo mettere in pausa il videoregistratore, per permettere alla telecamera di trovare il pezzo da riversare; la stessa cosa avveniva quando si doveva ricominciare a registrare.

IL DATABASE

Come si può osservare nella foto 4, il programma permette di creare un vero e proprio database video e di accedere alla libreria e alle clip memorizzate.
La Gold Disk non fornisce informazioni sulla capacità del database: abbiamo provato ad inserire i dati su due cassette, la prima con 5 clip e la seconda con 2; il file salvato su disco occupava solo 320 byte.
Se si considera il fatto che un floppy disk ha una capacità di 856.928 byte, ci si rende immediatamente conto della mole di informazioni che è possibile memorizzare, almeno in linea teorica.
Oltre alle normali funzioni di ordinamento, tipiche di un database (ordinamento alfabetico dei nastri e dei tagli, oppure secondo il contatore di partenza dei singoli spezzoni), un menu permette di modificare i parametri di uno spezzone, di cancellarlo, di cambiarne il nome o di visualizzarlo sul televisore. Tra le altre opzioni di rilievo, c'è la possibilità di inviare a una stampante l'elenco completo del database e della lista di montaggio; quella di salvare su dischetto in formato ASCII la lista degli eventi o di caricarne una creata con un word processor, per mandarla poi in esecuzione.

CONCLUSIONI

Se si riesce a utilizzare il sistema in modo automatico, tenendo conto del prezzo (389.000 Lire) e delle prestazioni offerte, si può sicuramente affermare che Video Director batte ogni concorrente.
Le centraline video dedicate al montaggio amatoriale hanno prezzi ben più elevati e possibilità molto più limitate; ad esempio, la Sony RM-E700 costa 1.000.000 di Lire e permette di montare, nell'ordine desiderato 20 scene al massimo; la Panasonic VW-EC310 costa 750.000 Lire e permette di montarne solo 8.
Se invece si utilizza il sistema in modo semi-automatico, Video Director è adatto a chi effettua montaggi video solo saltuariamente.
Si rimane invece molto perplessi di fronte all'uso di Video Director in modo manuale (in pratica rimane solo un database che indica quando azionare i due componenti video collegati e tutto l'hardware rimane inutilizzato), a meno che la Gold Disk non distribuisca solo il software a un prezzo più contenuto.
Il giudizio complessivo è in ogni caso positivo, tenendo conto sia della qualità del software sia del fatto che è il primo prodotto di questo tipo commercializzato in Italia. ▲

SCHEDA PRODOTTO

Nome Prodotto: Video Director
Casa Prodruttrice: Gold Disk (USA)
Distribuito da: SONDWARE Srl - Viale Aguggiari 62/a - 21100 Varese - Tel. 0332/232670
Prezzo: Lire 389.000
Giudizio: buono
Configurazione richiesta: Amiga 500 inespanso più varie attrezzature video (vedi articolo)
Pro: software di qualità.
Contro: costoso se si utilizza in modo manuale
Configurazione della prova: Amiga 500 1.3, 2 Mb, hard disk 20 Mb più varie attrezzature video (vedi articolo)

Digito

L'ultima volta abbiamo esaminato i comandi di movimento di Edit, ora cominceremo analizzando una variante dei comandi di movimento: le funzioni di ricerca. E' possibile comandare a Edit di muoversi avanti o indietro nel file fino a raggiungere la linea che contiene una determinata stringa. Se la stringa non viene trovata e la ricerca avviene in avanti, Edit arriverà fino alla fine del file sorgente e poi emetterà il messaggio "Input exhausted"; nel caso, invece, che la ricerca avvenga all'indietro, Edit si muoverà fino alla prima linea della coda di output e poi emetterà il messaggio "No more previous line". Se la stringa viene trovata, verrà visualizzata la linea che la contiene, che diventerà la nuova linea corrente. Se la linea corrente contiene già la stringa che stiamo cercando, Edit non farà nulla. I comandi sono: F (per Find), ricerca in avanti, BF per (Back Find), ricerca all'indietro.
La stringa da cercare va indicata subito dopo il comando (eventualmente separata da uno spazio). Per segnalare l'inizio e la fine della stringa si deve usare un delimitatore scelto arbitrariamente fra i seguenti:

```
/ . + - , ? : *
```

così, per esempio:

```
/pippo/   .pippo.   +pippo+
:pippo:
```

sono equivalenti tra loro. Bisogna solo evitare che il carattere delimitatore compaia all'interno della stringa di ricerca. Le stringhe possono avere anche dei qualificatori, che servono a limitare la ricerca a un determinato tipo di stringa. I qualificatori devono precedere immediatamente il primo delimitatore. Il qualificatore B (sta per Begin, inizio) indica che la stringa deve comparire a inizio linea; E (End, fine), che la stringa deve trovarsi alla fine di una linea; U, che la ricerca non deve tener conto di maiuscole e minuscole; P (Precisely, esattamente), che la linea deve contenere solo la stringa indicata, mentre L (Left, sinistra) ordina a Edit di cominciare la ricerca a partire dalla fine della linea e di procedere verso sinistra. I qualificatori possono anche essere combinati tra loro, anche se non in tutti i modi possibili; va anche notato che alcuni comandi non accettano certi tipi di delimitatori. Facciamo alcuni esempi:

```
F /pippo/
F U/pippo/
F PU/pippo/
```

Un caso particolare di stringa è quella nulla, che si indica con due delimitatori posti uno di seguito all'altro. Per trovare una linea vuota si può usare:

```
F P//
```

Per definizione, ogni linea è preceduta e seguita da una stringa nulla; così:

```
F E//  o  F L//
```

significano "vai alla fine di questa linea". mentre:

```
F B//  o  F //
```

ci posizionano all'inizio della linea corrente. Possiamo ora cominciare ad analizzare i comandi che modificano il file. In realtà, di uno abbiamo già parlato la scorsavolta: è W. Se vi ricordate, la funzione di W è quella di mandare in output tutto ciò che resta del file sorgente: Edit, però, mentre esegue questo lavoro, elimina automaticamente tutti gli spazi a fine riga. Quindi, caricando un file e impartendo il semplice comando W, potremo affrontare e risolvere questo compito, piccolo ma fastidioso, specie se lo si deve eseguire manualmente. Se, per qualsiasi motivo, volessimo evitare la rimozione degli spazi a fine riga, dovremo impartire il comando TR-; per ripristinarla, invece, basterà il comando TR+. L'ideale, però, sarebbe poter eseguire l'operazione di rimozione degli spazi con un solo comando CLI e non mediante una serie di comandi interattivi. Edit lo permette, mediante gli script. Uno script è un file di testo (ASCII) che può essere creato con qualsiasi text editor (come Ed). Nel testo andranno inseriti i comandi che Edit dovrà eseguire, con la stessa sintassi utilizzata in modo interattivo: in questo caso basterà il semplice carattere W. Salviamo poi il file con il nome "nospazi" nella directory S:. E' tutto: d'ora in poi, ogni volta che vorremo eliminare gli spazi a fine riga da un file di testo, digiteremo:

```
EDIT pippo WITH s:nospazi
```

e il file "pippo" sarà modificato secondo i nostri desideri. Per usare Edit è meglio crearsi degli script, quando possibile: sono molto più

semplici da usare dei comandi diretti e si evitano, a questo modo, errori che potrebbero farci perdere tempo o dati preziosi. Ma vediamo ora i comandi che modificano i file; ce ne sono di tre tipi: globali, per linea, per porzioni di linea. I più potenti sono i comandi globali: operano su tutto il file, linea dopo linea, man mano che questo scorre dal file di input verso quello di output. Una volta impartiti, rimangono sempre attivi, operando su ogni linea corrente. E' dunque necessario impartirli una sola volta, e poi scorrere il file fino alla fine con W per trasformare l'intero file. I comandi globali sono tre e iniziano tutti con G: il più utile è forse GE (per Global Edit) che sostituisce una stringa con un'altra e corrisponde praticamente ai comandi di ricerca e sostituzione dei normali text editor. Vediamo come si usa:

```
GE /pippo/pluto/
```

La stringa da cercare è "pippo", quella che dovrà prendere il suo posto è "pluto". Come si può notare il delimitatore finale della prima stringa costituisce anche il delimitatore iniziale della seconda. Questo vale per tutti i comandi che necessitano di due stringhe come parametri. Ora useremo GE per creare uno script che può risultare molto utile: aprite il vostro text editor preferito e inserite queste linee:

```
GE /o'/ò/
GE / pò/ po'/
GE B/pò/po'/
GE / e'/ è/
GE B/e'/è/
GE /e'/é/
GE /a'/à/
GE /u'/ù/
GE /i'/ì/
W
```

Salvate il testo con il nome "S:accenti" e avrete a disposizione un file script che trasforma le lettere seguite da un apostrofo in lettere accentate. Lo script contiene solo comandi GE, oltre al solito W. Man mano che Edit scorrerà il file, applicherà i comandi globali indicati ad ogni linea. Si noti che la prima linea sostituisce tutte le "o" con apostrofo con "o" accentate.
La seconda, invece, rimette l'apostrofo a "po'", come è giusto che sia nella lingua italiana.
La prima lo toglie e la seconda lo rimette: Edit, infatti, nell'eseguire i comandi globali segue esattamente l'ordine con cui i comandi sono stati impartiti. Si noti la presenza di uno spazio prima delle due forme della parola "po" nella seconda linea: serve a sostituire l'accento solo in "po'" e non in tutte le parole che terminano per "po"; non deve invece esserci lo spazio finale, perché "po'" potrebbe presentarsi a fine linea e in tal caso non sarebbe seguito dallo spazio. La terza linea si preoccupa di gestire un caso particolare: quello in cui la parola "po'" compare a inizio linea. Si potrebbe estendere lo script per fargli gestire altri casi particolari della lingua italiana, oppure creare il file opposto, per eliminare gli accenti e sostituirli con apostrofi: non è difficile, basta invertire le due stringhe dei comandi GE ed eliminare la gestione del "po'". Tenete presente che sono permessi al massimo 10 comandi globali. Ora potrete sostituire gli apostrofi del file "pippo" con il semplice comando CLI:

```
EDIT pippo WITH S:accenti
```

Il secondo comando globale è GA (per Global After), facciamo un esempio

```
GA \pippo\.ext\
```

Edit ricercherà tutte le occorrenze della stringa "pippo" e gli aggiungerà in coda la stringa ".ext", creando la stringa "pippo.ext". E' come sostituire la stringa "pippo" con "pippo.ext".
Il comando opposto è GB (per Before): la seconda stringa viene posta davanti a tutte le occorrenze della prima:

```
GB \pippo\DF0:\
```

trasformerà i vari "pippo" in "DF0:pippo".
Se volete sapere quali siano i comandi globali attivi, digitate SHG (SHow Global, mostra globali): apparirà l'elenco completo.
Se volete disabilitare un comando globale potete usare "DG numero", per riabilitarlo usate "EG numero", per cancellarlo usate "CG numero". In tutti e tre i casi, se si omette il numero, il comando verrà applicato a tutti i comandi globali esistenti.

Delimitatori di stringa
*/.+-,?:**

Qualificatori di stringa

B	a inizio linea
E	afine linea
U	non badare alle maiuscole
L	a partire dalla fine
P	esattamente

Varie

TR-	abolisci la rimozione degli spazi
TR+	ripristina la rimozione degli spazi
V-	abolisci la visualizzazione automatica
V+	ripristina la visualizzazione automatica

Movimento

N	vai alla linea successiva
P	vai alla linea precedente
M n	vai alla linea n
M-	vai all'inizio della coda di output
M+	vai alla fine della coda di output

Conclusione

W	vai alla fine del file e concludi
Q	come W
STOP	interrompi la sessione

Visualiuazione

?	visualizza la linea corrente
!	come ? con caratteri di controllo
T	stampa fino alla fine del file
T n	stampa n linee
TL	come T con numeri di linea
TL n	come T con numeri di linea
TN	stampa tante linee quante ne contiene la coda di otuput
TP	stampa il contenuto della coda di output

Ricerca per linee

F	in avanti
BF	all'indietro

Comandi Globali

GE	sostituzione
GB	aggiungi prima
GA	aggiungi dopo
DG	disabilita tutti i globali
DG n	disabilita il globale n
EG	abilita tutti i globali
EG n	abilita il globale n
CG	cancella tutti i globali
CG n	cancella il globale n
SHG	mostra i comandi globali

I comandi di Edit esaminati finora.

AREXX

Romano Tenca

Se la scorsa puntata è stata assimilata correttamente, l'interprete ARexx dovrebbe ora risiedere in memoria, in attesa di qualche programma da eseguire. Ha aspettato fin troppo. Come già dicevamo, un programma Arexx si può eseguire inviando alla porta REXX un messaggio particolare, che contenga il nome del file del programma. Il programma CLI Rx esegue proprio questo compito: accetta il nome di un file e lo comunica ad ARexx, che lo legge, lo decodifica e lo esegue:

```
Rx prova
```

eseguirà il programma contenuto nel file "prova". Dove viene cercato il file? Se il nome comprende un path, il file viene cercato solo nel path indicato, altrimenti nella directory corrente e poi nella directory REXX:. Sta all'utente definire mediante Assign la directory logica REXX. Se ARexx non trova il file "prova", tenta anche di trovare il file "prova.rexx". Aggiunge cioè al nome indicato l'estensione ".rexx", questo vale per entrambe le directory (in tutto, i tentativi sono quattro).
Rx ha però anche una seconda funzione. Invece di inviare ad ARexx il nome di un file, può inviargli una stringa, che ARexx esegue direttamente, come se fosse un programma (chiameremo questo tipo di programmi "programmi-stringa"). Per attivare questo modo, occorre usare la seguente sintassi:

```
Rx "corpo del programma"
```

Il segreto è tutto nelle virgolette (l'ultima non è neanche necessaria): la loro presenza segnala a Rx che ciò che segue sulla linea di comando non è il nome di un file, ma il corpo di un programma da eseguire direttamente. Questo modo è molto utile: in particolare, consiglio il suo utilizzo per familiarizzarsi con la sintassi dei comandi e delle funzioni ARexx. Non abbiate paura di provare: è veramente difficile (con i comandi standard) che un errore di sintassi possa causare un crash della macchina e la conseguente "Guru meditation"; potreste al massimo corrompere il file su cui state facendo delle prove, per cui è meglio lavorare su copie, magari in RAM.
La volta scorsa, se vi ricordate, abbiamo usato un programma-stringa all'interno dalla startup-sequence, per inizializzare ARexx.

CREIAMO UN PROGRAMMA

Un programma ARexx è un file di testo che può essere creato con qualsiasi editor (come Ed). Tutti i programmi devono iniziare con un commento, che ha la stessa sintassi dei commenti in linguaggio C:

```
/* sono un commento */
```

se il programma non inizia con un commento, non può essere riconosciuto come programma da Arexx (il commento non è però necessario nei programmi-stringa) e Rx visualizzerà questo messaggio d'errore:

Command returned 5/1: Program not found

Il messaggio è un po' ingannevole: è lo stesso che viene visualizzato quando il file non esiste; nel nostro caso, il file esiste ma manca il commento iniziale. Nel commento iniziale vale la pena di inserire il nome del file e magari altre informazioni relative al programma.
Visto che parliamo di commenti, ricordiamo che possono occupare più righe:

```
/* prima riga
   seconda riga */
```

Inoltre, possono essere nidificati: cioè un commento può essere contenuto in un altro:

```
/* prova /* sono un
commento nidificato */ */
```

Ciò è molto utile nei casi si voglia eliminare temporaneamente una sezione di codice da un programma e questa sezione contenga dei commenti.
Chiuso il primo commento, scriviamo il programma vero e proprio:

```
SAY "Hello world"
```

Scritte queste righe nel nostro text editor, possiamo salvare il tutto con il nome HELLO e poi eseguire il file con RX HELLO. Sulla console della shell apparirà il messaggio "Hello World".

UNA PRIMA ANALISI

SAY è un'istruzione del linguaggio ARexx che invia il parametro indicato sul resto della linea (una stringa, in questo caso) verso il canale di output corrente (la finestra della

console da cui si lancia Rx) aggiungendo una NEWLINE (il carattere 10). E' molto importante distinguere le istruzioni da tutti gli altri elementi del linguaggio, per evitare classici errori che vedremo a suo tempo. Molti linguaggi riconoscono le istruzioni fondandosi essenzialmente sul nome. Hanno una lista di parole riservate e, quando incontrano una parola che è presente in tale lista, la interpretano come un'istruzione. Non è questo il caso di ARexx. Tale linguaggio non ha una lista di parole riservate: le istruzioni vengono riconosciute in base alla posizione che assumono all'interno della linea. Osservate il nostro esempio: SAY appare come prima parola della linea (spazi e TAB iniziali vengono ignorati), NON è racchiusa tra virgolette e NON è seguita dal carattere "=". Quando ARexx incontra una linea così, cerca la prima parola (SAY) nell'elenco interno delle istruzioni e, se la trova, la esegue. Se non la trova, accadono altri eventi che, per ora, non ci interessano. Quindi, è la posizione a determinare se una parola debba o meno essere interpretata come un'istruzione e non il nome. Questa è una caratteristica fondamentale di ARexx che lo distingue da molti linguaggi e gli consente, come avremo modo di vedere, una flessibilità davvero unica. Avremmo potuto scrivere anche "say" invece di "SAY": prima di interpretare la linea, ARexx converte tutto ciò che non è compreso tra virgolette da minuscolo a maiuscolo. Bisogna tenere presente questa caratteristica, che non è importante per le istruzioni, ma per altri elementi del linguaggio, come vedremo tra poco. SAY accetta un solo parametro: un'espressione. Un'espressione è una sequenza di parole che può comprendere: costanti, simboli fissi o variabili, funzioni, operatori. Sono elementi classici di qualsiasi linguaggio e non sarà difficile al lettore riconoscerne il significato. Nel corso di questi articoli tratteremo comunque tali elementi uno per uno, al fine di chiarire come siano stati implementati in ARexx. Il contenuto dell'espressione viene valutato, cioè interpretato da ARexx, e il risultato diventa il parametro passato all'istruzione SAY. L'espressione da noi usata è composta unicamente da una stringa ("Hello World"), vediamo dunque cosa c'è da sapere sul loro uso.

LE COSTANTI STRINGA

Per segnalare ad ARexx che una certa sequenza di caratteri è una stringa occorre inserirla tra virgolette o apici:

```
 'sono una stringa'
 "anch'io"
```

La lingua italiana prevede l'uso dell'apostrofo, per cui sarebbe meglio evitare gli apici e ricorrere sempre o quasi alle virgolette doppie. Nel nostro secondo esempio, la stringa sarebbe finita dopo l'"h", se non avessimo usato le virgolette doppie. Se volessimo che all'interno della stringa risultante apparisse una virgoletta o un apice, potremo usare più metodi; provate a scrivere un programma come questo:

```
 /* Prova2 */
 SAY "apice tra virgolette
' "
 SAY 'vlrgoletta tra apici
"
 SAY 'apice tra apici '''
 SAY "virgoletta tra
virgolette """
```

negli ultimi due casi, l'apice e la virgoletta, che devono comparire nel corpo della stringa, sono stati raddoppiati, ma nell'output generato dal programma appaiono una sola volta; si tratta infatti di un modo per dire ad ARexx di non interpretare la virgoletta come segnale di chiusura della stringa. La gestione delle virgolette si complica quando si usa Rx per i programmi-stringa. Prima di tutto, come abbiamo già detto, Rx ha bisogno di una virgoletta iniziale per riconoscere la linea come testo di un programma: se vogliamo passare delle virgolette ad ARexx dobbiamo dunque usare le virgolette doppie:

```
Rx "say ""Hello World"" "
```

se però volessimo fare apparire delle virgolette entro le virgolette, non basterebbe più raddoppiarle, ma dovremmo a questo punto quadruplicarle:

```
Rx "say fuori "" dentro
"""" ancora """" "" "
```

Se volessimo scendere di un ulteriore livello dovremmo raggiungere le 8 virgolette e poi le 16 e così via... Tutto ciò, ovviamente, rende piuttosto scomodo l'uso di Rx in certe situazioni. Torniamo ora al programma iniziale, ricarichiamolo nel text editor e proviamo a modificare il testo in questo modo:

```
say Hello World
```

Abbiamo semplicemente eliminato le virgolette. Salviamo il programma ed eseguiamolo: il risultato sarà la stampa del messaggio HELLO WORLD. La stringa è diventata tutta maiuscola (poco male, c'era da aspettarselo), ma il fatto grave è che Hello World non è più una stringa, anche se lo sembra: per ARexx si è trasformato in due variabili accostate (lo si capirà meglio quando tratteremo delle variabili). Anche questo è estremamente importante. Se in precedenza avessimo definito la variabile Hello con un altro valore, invece della stringa HELLO sarebbe stato stampato il valore della variabile. Inoltre, non trovando le virgolette, ARexx è costretto a guardare nell'elenco delle proprie variabili per vedere se ne compare una di nome Hello, cosa che porta via una certa quantità di tempo, anche se minima. E' quindi una buona regola di programmazione porre sempre le costanti stringa tra virgolette, anche quando non sembrerebbe strettamente necessario. Potrebbe diventarlo in futuro, quando magari modificheremo il codice, aggiungendo una nuova variabile. In ogni caso risparmieremo del tempo prezioso. ARexx ha tantissimi pregi, ma uno gli manca di sicuro: la velocità. Nelle stringhe possono comparire anche

caratteri non ASCII, per esempio le lettere accentate, che non sono permesse al di fuori delle virgolette. Se uno di questi caratteri viene usato, per esempio, come nome di variabile, ARexx segnalerà un errore del tipo:

```
+++ Error 8 in line 4:
Unrecognized token
 Command returned 10/8:
Unrecognized token
```

"Token" è il termine generico con cui ARexx indica gli elementi del linguaggio, siano essi nomi di variabile, di istruzione, di funzione e così via.
Se vogliamo inserire dei caratteri di controllo in una stringa, possiamo indicare direttamente il valore esadecimale dei caratteri mediante questo costrutto:

```
'0A'x
"0A"x
```

0A verrà trasformato nel carattere di NEWLINE (10 decimale). Si possono porre più caratteri in sequenza:

```
'0A0D'x
```

e anche inserire spazi per una maggiore leggibilità; verranno del tutto ignorati dall'interprete:

```
'0000 0004'x
```

In questo caso, avremo tre caratteri 0 seguiti da 4.
Si noti che questo tipo di costrutto può anche essere concepito come una longword che contiene il valore decimale 4. Alcuni funzioni di ARexx possono aver bisogno di tale tipo di parametro. Però ai fini dell'interprete, tale longword appare a tutti gli effetti come una stringa di 4 caratteri. La x che segue il secondo apice indica che la stringa contiene dei valori esadecimali; se invece della x si usasse una b, la stringa dovrebbe contenere dei valori binari:

```
"01001000" b
```

questo corrisponde al valore esadecimale 48 (decimale 72, al carattere H maiuscolo in ASCII).
Infine, ricordo che una stringa è come un commento: non finisce finché non viene chiusa. Pertanto è sempre possibile andare a capo:

```
SAY "
Hello Wor
ld"
```

darà il solito messaggio (gli a capo sono eliminati dall'interprete), come se fosse stato scritto su una sola linea. L'argomento trattato sembrava semplice, in realtà ci ha tenuti piuttosto occupati, ma è di estrema importanza perché ARexx gestisce praticamente ogni cosa mediante le stringhe, anche i numeri, come vedremo la prossima volta. ▲

Silvio Umberto Zanzi

Dopo la trattazione estremamente teorica della scorsa puntata, torniamo su qualcosa di più pratico: la gestione della stampante attraverso un programma in C. Il compito che ci prefiggiamo non è per nulla complicato, bene o male chiunque ha avuto la possibilità di vedere all'opera una di queste periferiche, non vi sono concetti astratti da assimilare e non vi è neppure la necessità di ricorrere a strane librerie, accedere ai chip custom o fare "acrobazie" con i puntatori, quindi rilassatevi! Se tralasciamo il fattore prezzo, possiamo differenziare le stampanti attraverso la tecnologia di funzionamento, possiamo, ad esempio, citare la stampa tramite aghi o una margherita, o mediante il fissaggio di toner attraverso un laser, per getto di inchiostro liquido, per trasferimento magnetico di inchiostro termosensibile in polvere, ecc. Un principiante potrebbe, in un primo momento, credere che un programma, che in qualche modo debba stampare, abbia la necessità di riconoscere la stampante in linea e adattarsi alla sua specifica tecnologia, fortunatamente le cose non stanno così. Per prima cosa esistono le preferences: con questo programma l'utente rende noto ad Amiga ogni caratteristica della propria stampante, segnala, in pratica, se la propria stampante è una normale 9 aghi collegata alla porta parallela oppure una sofisticata laser collegata alla porta seriale, se è in grado di stampare a colori, se è 80 o 132 colonne, se funziona a modulo continuo o a frizione e via dicendo. Come secondo punto, Amiga dispone di un nome simbolico univoco tramite il quale accedere alla stampante; allo stesso modo con cui DFO: si riferisce al primo drive del computer, PRT: si riferisce all'unità di stampa e con la stessa facilità con cui si copia un file di testo sul primo drive è anche possibile stampare un file ASCII. Chiunque abbia digitato da CLI la sequenza Copy Nome-file-testo PRT: sa cosa intendo. Da un punto di vista pratico, un programma ha spesso la necessità di dover stampare una serie di dati che ha generato in memoria, considerate, ad esempio, tutti i programmi per l'elaborazione dei testi, i fogli elettronici, i pacchetti statistici, la miriade di monitor/analizzatori/manipolatori di pubblico dominio, ecc. Vediamo come si può realizzare una stampa, sviluppando, con un breve programma C, l'idea che ho appena descritto.

IL PROGRAMMA

Dopo aver incluso gli header file necessari, definiamo alcune variabili. Avremo bisogno di un array di char per memorizzare il testo da stampare, di una variabile di tipo char da usare come buffer e di un contatore.

```
char Memoria [135];
char carattere;
int contatore=0;
```

Avremo anche bisogno di due variabili per contenere ciò che si definisce handler di un file. L'handler non è altro che un puntatore ad un determinato file, puntatore che si ottiene passando alla funzione DOS Open() il nome del file stesso e le sue modalità d'accesso.

```
BPTR PrtHandler,
FileHandler;
```

Osservando questa definizione, notiamo un tipo di dato mai usato in nessun programma precedente, il BPTR, abbreviazione di BCPL POINTER. Il BCPL è un linguaggio di programmazione che ha avuto il suo periodo di massima diffusione attorno ai primi anni '70, è il progenitore del C e come tale ha alcune caratteristiche in comune, ma anche molte differenze rilevanti. Un confronto fra il C e il BCPL esulerebbe dallo scopo di questo articolo, ci limiteremo soltanto ad analizzare i tipi di dati, le cui differenze hanno un notevole peso nella programmazione Amiga. In C, come ben sappiamo, esistono tipi di varie dimensioni e una discreta serie di qualificatori per estendere o limitare i tipi di base. Il BCPL, invece, è un linguaggio senza tipi e permette solo manipolazioni a 32 bit. Ora consideriamo il 68000, questo versatile processore lavora su indirizzi di memoria a 32 bit, ma permette riferimenti ad esso solo con offset di byte. Questa caratteristica non crea nessun problema quando si usano i puntatori C: il compilatore conosce le caratteristiche di Amiga e genera del codice con gli adeguati indirizzamenti. In BCPL, invece, questo non è possibile, non vi è modo di specificare offset a byte, i puntatori, quindi, non sono compatibili. Quando si ottiene un puntatore come ritorno da una funzione DOS, e vi è la necessità di passarlo ad una qualsiasi funzione

del resto del sistema operativo, bisogna sempre eseguire una conversione moltiplicando il BPTR per quattro (e dividendolo per quattro quando se si esegue l'operazione inversa). Molti di voi a questo punto si sentiranno confusi, ciò è naturale soprattutto se non avete esperienze di microprocessori, quindi non preoccupatevi se non siete riusciti a seguire il discorso sui puntatori BCPL. Sappiate solo che ogni volta che dovete passare un normale puntatore a una funzione DOS, o viceversa, DOVETE eseguire SEMPRE una conversione secondo le seguenti regole:

Manipolazione	Conversione richiesta
C -> BCPL	puntatore C / 4
BCPL -> C	puntatore BCPL * 4

Nasce spontaneo chiedersi perché è stata realizzata una parte di Amiga in BCPL, mentre tutto il resto è stato fatto in un misto di C e assembler.
La risposta ricade nella leggenda: i creatori di Amiga avevano quasi ultimato la realizzazione del sistema operativo, avevano creato tutte le sezioni principali, mancava soltanto la parte riguardante il DOS. Purtroppo i progettisti erano molto in ritardo sulla tabella di marcia prevista, avevano delle date da rispettare e mancava il tempo materiale per creare un DOS daccapo. Decisero, allora, di adottare un sistema operativo già esistente, realizzato all'università di Cambridge, denominato TRIPOS. Quest'ultimo si adattava bene alle strutture di Exec e al concetto di Device, e, nonostante fosse scritto in BCPL, fu scelto e implementato su Amiga in circa tre settimane. Per questa necessità puramente commerciale, ogni volta che si usa la libreria DOS bisogna seguire alcune accortezze di interfacciamento tra C e BCPL. Esiste un testo molto interessante in lingua inglese sull'implementazione del BCPL in Amiga, corredato da esempi in C, note storiche, bibliografia e informazioni sul TRIPOS. Tale testo è di pubblico dominio e i possessori di modem lo possono trovare su molte BBS, ad esempio, su IDCMP BBS di Imola (Tel. 0542-25983 300-14400HST Bps) si trova in area file 11 con il nome di BCPLSTUF.ARJ.
Torniamo alla trattazione tecnica, abbiamo definito due puntatori BCPL atti a contenere gli Handler, ora dobbiamo ottenere l'handler per il file di testo da leggere.

```
if ((FileHandler = Open
("ram:TestoProva",
MODE_OLDFILE)) == NULL)

Exit(NULL);
```

La funzione Open richiede come argomenti il nome del file e le sue modalità d'accesso, quest'ultime vengono identificate tramite alcune macro definite nell'include file Libraries/Dos.h

Modalità	Azione
MODE–OLDFILE	Il file deve già esistere
MODE–NEWFILE	Viene creato un nuovo file
MODE–READWRITE	Se il file non esiste, il DOS lo crea automatica mente

La funziona ritorna l'Handler come puntatore BCPL oppure zero se fallisce, un tipico fallimento può essere quello generato dal passaggio alla funzione di un nome di file, con modalità MODE–OLDFILE, non esistente sul disco.
Nell'esempio qui illustrato ho inserito come file da manipolare Ram:TestoProva, tale file (presente sul disco assieme a questo esempio) dovrà essere copiato in RAM Disk. Open() non potrà generare l'-Handler se non troverà il file in RAM:, questo perché abbiamo specificato come modalità MODE–OLDFILE, inoltre, in tal caso, Open() ritornerà zero e il programma uscirà immediatamente usando la funzione DOS Exit(). Se avessimo invece specificato la modalità MODE–NEWFILE, senza copiare il file in RAM:, un nuovo file di nome TestoProva sarebbe stato automaticamente creato.
Ora vogliamo copiare tutto ciò che è presente nel file TestoProva in memoria, per fare questo useremo una funzione di lettura in un ciclo for:

```
for (contatore=0;
contatore < 135;
Read(FileHandler, &carattere,1))
    I
  Memoria[contatore]=carattere;
  printf("%c",Memoria[contatore]);
     ++contatore;
```

Qui vediamo l'uso di una nuova funzione: Read, così definita:

```
Read (handler_file,
buffer, lunghezza)
```

Handler_file è l'handler del file da leggere, lunghezza è il numero di byte da leggere, mentre buffer è l'indirizzo dove copiare le informazioni lette. La funzione ritorna il numero di byte letti, valore molto utile per eseguire controlli. Se, ad esempio, il valore ritornato è inferiore a quello richiesto in lunghezza, significa che il file era più corto del previsto e ciò può causare dei problemi.
Nel nostro esempio specifico viene letto un carattere per volta (lunghezza=1) dal file RAM:TestoProva, questo viene memorizzato in un array e mostrato contemporaneamente sul video.
Il ciclo è stato fatto in modo da soddisfare la lunghezza del file TestoProva presente sul disco, se notate, infatti, non vengono eseguiti controlli.
Concluso il ciclo, l'array sarà una copia esatta del file TestoProva e potrà essere stampato, dopodiché l'handler di TestoProva non avrà più nessuna utilità e potrà essere rilasciato con la funzione Close().

```
Close(FileHandler);
```

Per questo mese è tutto, ricordiamo che i file d'esempio sono contenuti nel cassetto BasicFacile.

PROGRAMMAZIONE FACILE IN BASIC

Domenico Pavone

Uno degli aspetti più rilevanti della programmazione, qualunque sia il linguaggio adottato, è rappresentato dal rapporto tra il computer e i dispositivi esterni ad esso collegati, in alcuni casi di vitale importanza: floppy e hard disk, stampante, video, eccetera.
AmigaBasic consente la gestione di queste periferiche in modo abbastanza semplice, come largamente illustrato nella sua manualistica a proposito della manipolazione dei file su disco.
Lo stesso meccanismo, con modalità più o meno particolari, può d'altra parte essere applicato anche per esigenze diverse, coinvolgendo l'uso di *device* forse meno sfruttati, ma estremamente utili in alcune circostanze.
Intanto, generalizzando, cominciamo col considerare la generica procedura di accesso alle periferiche, che può essere riassunta in tre passi: apertura di un canale di comunicazione con il dispositivo, invio o ricezione di dati, e, infine, chiusura del canale di comunicazione. In pratica proprio lo stesso tipo di procedura comunemente adottato per il trattamento dei file-disco, ma, come ovvio, *citando* differenti nomi di device nella sintassi dei comandi preposti ad assolvere i tre compiti prima accenati.
Rimandando alla lettura del manuale di AmigaBasic e ai trascorsi appuntamenti con questa rubrica per una più generale cognizione dell'argomento, in questa sede ci occuperemo di due di questi d*evice*, CON: e COM1:, pressoché ignorati dalla manualistica a dispetto della loro utilità.
Con il primo, in particolare, è possibile supplire a una carenza di AmigaBasic in rapporto all'output di testi sullo schermo.
E' noto, infatti, come sia impossibile da programma associare attributi di stile (neretto, sottolineato, corsivo) al testo, se non ricorrendo a librerie di sistema non sempre di facile comprensione. L'assenza di comandi dedicati può tuttavia essere aggirata con maggiore facilità trattando lo schermo esattamente come si farebbe con una stampante, ovvero inviando a quel d*evice* una sequenza di caratteri speciali che attivi l'effetto desiderato. Per chi non l'avesse ancora capito, si sta parlando del cosiddetto standard ANSI, quello stesso direttamente sfruttabile da ambiente DOS. Per prenderne atto, si provi ad aprire una finestra Shell e, dopo avere premuto una volta il tasto Escape, a digitare al suo interno i caratteri **[1m**.
Il testo, da questo momento in poi, apparirà in neretto. Se volessimo ottenere, da BASIC, lo stesso effetto su un testo su carta, basterebbe inviare la stessa sequenza alla stampante, adoperando in questo caso il device PRT:, che utilizza direttamente i settaggi delle Preferences per adattarsi alle caratteristiche della stampante. In termini pratici, quanto appena affermato si tradurrebbe in questa serie di istruzioni BASIC:

```
OPEN "PRT:" FOR OUTPUT AS
#1
PRINT #1,CHR$(27)+"[1m"
CLOSE 1
```

Il carattere Escape, come si può notare, è espresso dal suo corrispondente valore ASCII, che è appunto 27.
Già, ma come applicare lo stesso metodo a una finestra di AmigaBasic? Il manuale dell'interprete Microsoft cita l'esistenza di un device SCRN: riferito allo schermo, ma se si prova a sostituire nel mini listato, di cui sopra, la citazione SCRN: al posto di PRT:, si constaterà l'inefficacia della soluzione: verrà semplicemente riprodotta carattere per carattere la sequenza ANSI.
La soluzione, però, è a portata di mano. Nell'esperimento precedente si era adoperata una finestra del DOS, che notoriamente sfrutta il device CON:.
Basterà adoperare questo stesso device anche da BASIC, e con le stesse modalità, per ottenere quanto ci si era prefissato. Si provi, a titolo di esempio, ad attivare un listato come questo:

```
fil$="con:80/20/150/150/
Ansi test"
esc$=CHR$(27)+"["
deflt$=CHR$(27)+"[0m"
OPEN fil$ FOR APPEND AS #1
LEN=1
PRINT#1,esc$"1m";
"Grassetto"
PRINT #1,deflt$
PRINT#1,esc$"3m";
"Corsivo"
PRINT #1,deflt$
PRINT #1,esc$"4m";
"Sottolineato"
PRINT #1,deflt$
PRINT #1,esc$"1m"esc$"4m";
"Misto"
PRINT #1,deflt$
PRINT #1,esc$"7m";"
```

```
Reverse "
PRINT #1,deflt$
PRINT #1,esc$"42;33m Color
"
PRINT #1,deflt$
WHILE MOUSE(0)=0:WEND
CLOSE 1:END
```

Stavolta, in una finestra delle dimensioni specificate nella variabile *fil$*, come da analoga sintassi AmigaDos, verrano mostrati i vari stili di scrittura possibili. Per uscire dal programma occorrerà *cliccare* con il mouse nella finestra di output del BASIC, all'esterno di quella titolata Ansi Test.
Si noti, in particolare, come gli stili siano associabili: si potrà avere un neretto sottolineato, o un corsivo in grassetto (e così via) semplicemente inviando più sequenze ANSI al device CON:.
Oltre che gli stili di scrittura, l'esempio mostra anche come con questa tecnica possa essere modificato il rapporto colore tra sfondo e primo piano, pur se con maggiori limitazioni rispetto al tradizionale Color del BASIC.
Per un uso proficuo, basterà consultare una tavola delle sequenze ANSI (ISO) e scegliere quelle che servono ai propri scopi.

LA PORTA SERIALE

Se lo schermo, tutto sommato, può essere trattato come una qualunque altra periferica, non si può dire lo stesso del device COM1:, riservato a una gestione specializzata della porta seriale. Tramite COM1:, infatti, è possibile pilotare una tipica attività da *terminal*, perlopiù identificabile in un rapporto computer-modem.
Considerate le limitazioni legate alla presenza dell'interprete e la diffusione sul mercato di efficienti programmi di comunicazione, la progettazione di un terminal in BASIC può sembrare anacronistica. Tuttavia, anche senza obbligatoriamente dover prevedere la presenza di un modem (si pensi alla connessione fisica di due computer attraverso la porta seriale), è comunque possibile qualche applicazione personalizzata, per esempio volta ad automatizzare il collegamento con un modem *remoto*.
Da un punto di vista pratico, il device COM1: presenta qualche differenza applicativa. Intanto, la fase di apertura del dispositivo non richiede che si precisi la parola chiave Input, Output oppure Append. Data per scontata una anche superficiale consultazione del manuale di AmigaBasic, basterà qui accennare al necessario settaggio di alcuni parametri riguardanti la velocità di trasmissione/ricezione dei dati e il formato che dovranno assumere i byte che li contengono. Anche senza una precisa conoscenza del significato di questi parametri, in pratica occorrerà badare, soprattutto, a farli coincidere in entrambi i terminali interessati alla comunicazione, siano essi collegati via modem che via cavo seriale. Inoltre, proprio per un limite intrinseco del BASIC, è opportuno non settare velocità di trasmissione superiori a 2400 Baud.
Ma prima di aggiungere qualche opportuno dettaglio, ecco come, con pochissime righe di programma, è implementabile un terminale assolutamente funzionante finalizzato all'uso di un modem che supporti i 1200 Baud:

```
WIDTH 80
apre$="com1:1200,N,8,1"
OPEN apre$ AS #1
PRINT #1,"ATZ"
ON BREAK GOSUB fine
BREAK ON
WHILE 1
tasto$=INKEY$
IF tasto$<>"" THEN
PRINT #1,tasto$;
PRINT tasto$;
END IF
IF LOC(1)=1 THEN
PRINT INPUT$(1,1);
END IF
WEND
fine:
BREAK OFF:CLOSE 1:END
```

Volendolo adattare alla velocità immediatamente superiore, basterà modificare il valore 1200 contenuto nella variabie stringa apre$ in modo che diventi 2400. In generale, il suo funzionamento può così essere sintetizzato: ogni carattere digitato da tastiera viene inviato alla seriale da un normale Print# sul canale aperto allo scopo, mentre lo stesso carattere viene poi normalmente stampato a video, emulando il cosiddetto *full duplex*.
Per la fase di ricezione, invece, occorre un controllo sulla porta, affidato alla funzione del BASIC Loc(), che restituisce valore 1 se un byte è pronto per essere letto, oppure 0 in caso contrario.
Il terminale è, come ovvio, molto rudimentale, ma si presta a migliorie di ogni sorta.
Se si considera l'uso associato a un modem che accetti lo standard (ormai pressoché universale) Hayes, in pratica basterà digitare direttamente da tastiera (o creando appositi menu) i comandi accettati da questo standard per fare tutto ciò che si desidera. Così, una volta attivato il programma, si potrà, per esempio, effettuare una chiamata telefonica digitando ATDP seguito dal numero di telefono, o (come fa la quarta riga del listato) inviare una stringa che resetti la configurazione del modem.
Inutile dire che, per quanto l'uso di COM1: renda possibile quasi tutte le comuni applicazioni legate al modem, è anche necessaria una seppur minima conoscenza dei meccanismi d'azione di questa periferica, in modo da adattarne le esigenze al trattamento effettuato dal programma di comunicazione.
Il listato proposto si limita alla ricezione/invio dei dati al solo scopo di visualizzarli a video, ma nulla vieterebbe (teoricamente) che gli stessi venissero, per esempio, indirizzati a un file-disco aperto in scrittura, consentendo così la ricezione di file da un terminale all'altro con il modem a fare da tramite.
Ma, in simili casi, la sicurezza del trasferimento diventa un'esigenza imprescindibile, e il compito di progettare un algoritmo che la garantisca si fa decisamente complesso, tanto da scoraggiarne l'implementazione in BASIC. ▲

di Simone Crosignani

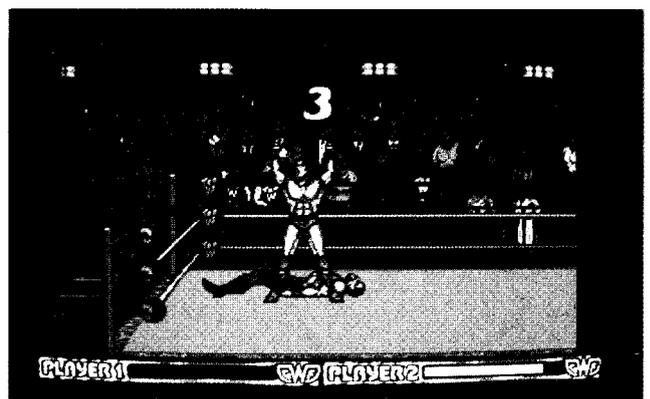
PROJECT X

TEAM 17

Rieccoli dopo lo strepitoso successo e i riconoscimenti ottenuti su tutte le riviste del settore con *Alien Breed* e *Full Contact,* il Team *17* torna all'attacco con un nuovo gioco A dire il vero siamo molto contenti dei risultati ottenuti da questi ragazzi ex-membri della scena Amiga (pirati N.d.R.) e in cui questa rivista ha sempre creduto, segnalandoveli già parecchi mesi fa, quando la concorrenza non ne consoceva nemmeno l'esistenza. Questo, modestia a parte, ci fa molto piacere, soprattutto dopo aver visto la loro ultima produzione, *Project X,* che può essere tranquillamente etichettato come il miglior shoot 'em up mai visto su Amiga. Questavolta Rico Holmes & C. hanno infatti deciso di mostrare la propria abilitàcon un gioco completamente diverso dalle loro precedenti produzioni: così, dopo un picchiaduro, un gioco di guida e un arcade con inquadratura dall'alto, il team (ormai indipendente al cento per cento) ha tentato di realizzare un classico per Amiga seguendo i canoni più tradizionali per uno sparatutto: scrolling orizzontale, armi extra, power up, beam, nemici di fine livello e tante, tante, tante astronavi a riempire lo schermo. Ma procediamo con ordine, dal menu principale. La prima cosa che si nota, oltra al logo gigantesco di Project X e alla colonna sonora techno rave davvero "pompatissima", è il cast d'eccezione: oltre a Holmes, Brimble e soci, segnaliamo Stefan Boberg (autore del magnifico LHA) come programmatore aggiunto, Bjiorn Lynne (colonna sonora di Fantastic Voyage) come consulente musicale e Rob Northern. Le opzioni non sono moltissime, ma hanno una discreta dose di originalità: per cominciare è possibile scegliere fra tre astronavi diverse, ognuna con differente potenza di fuoco e maneggevolezza. Si può poi cambiare la già citata musica rave con una melodia più orecchiabile da shoot'em up classico, salvare e caricare gli high score e selezionare il modo Arcade o Rookie: con il primo il gioco è più difficile, ma è possibile ripartire dai livelli più avanzati e scrivere il proprio nome nella tabella dei record, mentre il secondo è una variante facilitata e limitata. Settato il gioco a proprio piacimento, si parte con una schermata d'introduzione (diversa ad ogni livello) e con il gioco vero e proprio: come nel vecchissimo *Nemesis,* distruggendo le varie ondate di nemici, si ha la possibilità di raccogliere un power up che va ad aumentare l'arsenale a propria disposizione Tutto questo fino ad arrivare allo scontro con il nemico finale, più cattivo e coriaceo che mai Project X e, da ora, la nuova pietra di paragone nel campo degli shoot em up programmazione, grafica e sonoro non hanno nulla da invidiare a Z-Out e la giocabilità è persino superiore a quella di Silkworm. E' inutile che vi dica di andare a comprarlo... Un unico avvertimento: questo gioco è parecchio difficile, quindi non aspettatevi di portarlo a termine in pochi giorni! Uomo avvisato...

TOP WRESTLING

GENIAS

Il wrestling: una disciplina (o indisciplina?) sportiva di enorme successo negli USA, discretamente seguita anche nel nostro paese, come dimostra l'impegno preso da Telepiù 2 di trasmettere interamente in diretta Wrestlemania 8, il meeting più importante in questo campo (in America

gli spettatori collegati per quest'evento erano svariate decine di milioni!). Purtroppo fino ad oggi le simulazioni di wrestling per Amiga si erano rivelate, per la stragrande maggioranza, di bassissimo livello e il solo WWF Wrestlemania della Ocean, uscito a inizio anno, meritava di essere preso in considerazione. La Genias, reduce dallo strepitoso successo di Warm Up, ha deciso di sopperire a questa lacuna con il suo ultimo prodotto: Top Wrestling. Inutile dire che questo gioco parte con un discreto handicap rispetto al titolo Ocean per la mancanza dei nomi di richiamo della World Wrestling Federation, di cui la software house anglosassone detiene i diritti, ma questo in fondo è l'unico fattore a svantaggio di Top Wrestling. I sedici lottatori di cui potete disporre, infatti, non si chiamano Hulk Hogan o Ultimate Warrior, ma sono uguali a loro in tutto e per tutto. Oltre a scegliere il proprio personaggio, in Top Wrestling è possibile disputare un incontro singolo (da soli o in due contemporaneamente) o lanciarsi nel torneo per la conquista del titolo di campione del mondo. Dopo aver assistito a una paio di animazioni notevoli si passa così al match vero e proprio: le mosse possibili sono numerosissime e, nonostante vincere sia discretamente difficile, la giocabilità è parecchio superiore a quella di WWF Wrestlemania, la grafica è curata, e quel che più conta, le animazioni sono elevate in quantità e qualità. Proprio la grafica è il punto forte di Top Wrestling: oltre a quella dei vari lottatori sono da segnalare senz'altro le animazioni e le schermate varie che fanno da contorno al match, realizzate in puro "stile Cinemaware". Il sonoro è buono, la programmazione pure... Cosa volete di più?

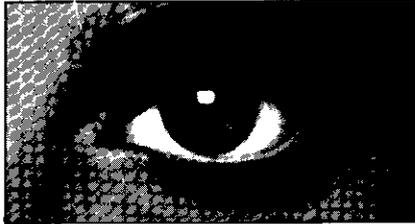
AGONY

PSYGNOSIS

Esistono giochi per Amiga che, per la cura con cui sono stati realizzati, vengono usati per parecchi anni come termine di paragone: così, ad esempio, quando un titolo è tecnicamente notevole, nasce spontaneo il confronto con Shadow of the Beast, uno dei migliori prodotti Amiga come programmazione, grafica e sonoro ancora oggi. E' quindi comprensibile lo stupore provato dopo le dichiarazioni degli "addetti ai lavori" della Psygnosis che indicavano Agony come il possibile "Beast degli shoot'em up". Dichiarazioni che solo ora, provando il gioco finito, possiamo confermare in pieno. Agony, infatti, non è un giocoo, per dirla meglio, si avvicina come concept molto più a un demo che a uno sparatutto vero e proprio. Com'è possibile? Semplice: in passato la software house di Liverpool ha sempre fatto incetta dei migliori programmatori disponibili sul mercato, e, fatta esclusione per gli stipendiati della Rainbow Arts, solo un team era stato in grado di produrre qualcosa di egualmente valido, pur lavorando per un'altra casa: stiamo parlando di Art & Magic, autori di Unreal per la Ubisoft. Così il signor Psygnosis ha deciso di reclutare lo squadrone francese e integrarlo con una decina di grafici e musicisti anglosassoni (il mitico Roger Dean ha curato il logo, come al solito, e Jerome Tel, ex-Maniac of Noise, s'è occupato della colonna sonora). Il risultato è tecnicamente mostruoso: la grafica è stupefacente (Sauer, che ha disegnato le schermate introduttive di ogni livello, non e di questo pianeta), il sonoro è magnifico e la programmazione è sugli stessi livelli. E il gioco? Scarsino... Essenzialmente si tratta di uno sparatutto a scrolling orizzontale con le solite armi extra e i classici nemici di fine livello e questa mancanza d'originalità si sente eccome: anche la giocabilità è piuttosto limitata, soprattutto a causa delle dimensioni eccessive del protagonista e di quelle limitate dell'area di gioco. Inoltre, Agony è, contrariamente alla maggioranza dei giochi Psygnosis, incredibilmente facile e finirlo non è certo impresa titanica, anche se l'unico stimolo a proseguire nei livelli è il desiderio di vedere nuovi mostri e nuove schermate grafiche. Realizzazione tecnica perfetta, giocabilità sottozero: a voi la scelta se comprarlo o no... Secondo noi se volete spendere saggiamente i vostri soldi sarebbe meglio optare per Project X, recensito in questo numero, molto più veloce e giocabile. Se invece volete una prova di quanto vale il vostro Amiga non c'è alcun dubbio che Agony è il titolo che fa per voi.

COSA HA 16 MILIONI DÌ COLORI - UN FRAME BUFFER A 24 BIT - GENLOCK - FRAMEGRABBLER - ELIMINATORE DI FLICKER - PICTURE IN PICTURE - TITOLATRICE VIDEO E UN SISTEMA DI MODELLING TRIDIMENSIONALE ?

Introduciamo la GVP IMPACT VISION 24

La periferica Video "All in one" per A3000 e A2000

Impact Vision 24 è un sogno che si avvera per la tua A2000 - A3000: E' la prima periferica multifunzione specificatamente pensata per gli slot di espansione dell'A3000. Con il kit opzionale "genlock slot" A2000 completa ed aumenta le performance deil'A2000. Guarda queste "features", tutte assieme in una "UNICA" scheda di espansione !!

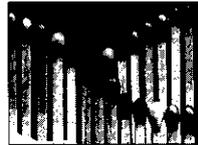
▶ **GENLOCKS (Rgb+Sync)**
Il Genlock Rgb opera digitalmente, per produzione di qualità assolutamente professionale: niente diffusioni di colori, niente false immagini, ecc.

▶ **1,5 MB Frame Buffer.**
Mostra a 24bit, 16 milioni di colori sul tuo monitor.
In un monitor Multi-Sync, puoi anche vedere 16 milioni di immagini a colori in modo non interlacciato.

▶ **Framegrabbler/Digitizer in tempo reale.**
Congela, blocca e salva (nei formati standard a 4096 col., oppure IFF 16M colors) qualsiasi immagine proveniente da sorgenti video RGB. Per "bloccare" in modo videocomposito o S-VHS è necessario un RGB splitter (optional).

▶ **Eliminatore di Flicker.**
Raddoppia e potenzia la circuiteria video dell'A3000.De-interlaccia qualsiasi video esterno!! Una necessità per tutti i possessori di A2000.

Chiedi il nostro programma A2000 "genlock slot trade up" se già usi un gen lock in modo insoddisfacente!!!

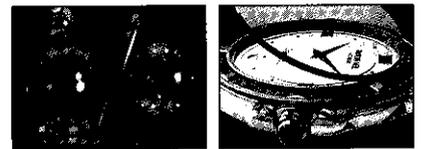
▶ **Uscite RGB, Videocomposito, e S-VHS simultanee.**
Ora qualsiasi cosa tu vedi su Amiga può essere registrato su videotape, incluse animazioni, immagini a 24bit etc.

▶ **Immagine nell'immagine (PIP).**
Congela, ridimensiona e/o riposiziona immagini RGB in movimento come qualsiasi altro "workbench", semplicemente con un doppio click del mouse o mediante una "hot key".
Con un monitor Multi-Sync tutto questo può avvenire anche in modo non interlacciato.
Possibile e "unico" anche il "Reverse PIP" che ti permette di vedere un workbench completamente funzionante o qualsiasi altra applicazione sullo schermo (con possibilità di ridimensionamento e riposizionamento) mentre "gira", a schermo pieno, un live video.

▶ Per assicurare un immediato sfruttamento delle performance della tua nuova IMPACT VISION 24 video station, includiamo in ogni prodotto alcuni software:

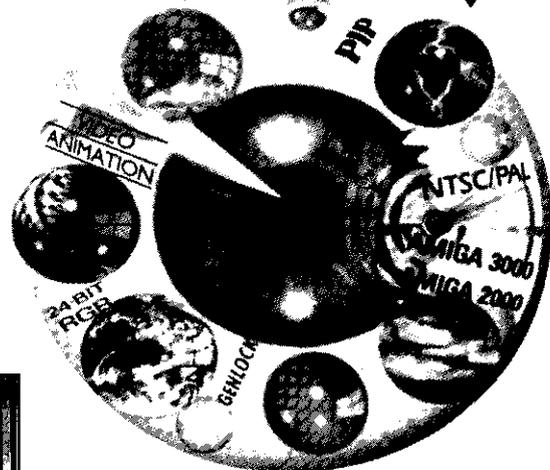
• **CALIGARI - IV24**
Una versione esclusiva di uno dei migliori software 3-D in qualità broadcast di disegno e modelling. Usa la tua immaginazione per modellare scene tridimensionali avendo a disposizione 16 M di colori.
Usa le tue immagini digitalizzate video come tessuti per raggiungere qualsiasi obiettivo! Il limite è la tua mente!!

• **SCALA - Titolatrice**
Facile da usare, questo pacchetto è completo di tantissimi fonts speciali e di tanti eccitanti effetti speciali.
Fai diventare il tuo Amiga un generatore di caratteri!!

• **MACROPAINT IV24**
Un programma di grafica con 16M di colori in 2D che ti permetterà di creare e manipolare qualsiasi coloratissima immagine a 24 bit.

• **Pannello di controllo.**
Permette un completo controllo software su tutte le numerose features dell'IMPACT VISION 24.

Puoi usare il mouse semplicemente o utilizzare una "hot key" configurabile per attivare qualsiasi cosa.

Alla GVP, abbiamo cercato di creare un grande entusiasmo agli utilizzatori "PRO" di A2000/ A3000.
Con IMPACT VISION 24 ce l'abbiamo fatta!!!

GVP

40057 CADRIANO (BOLOGNA)
VIA A. GRANDI, 22
TELEFONO. (051) 76.52.99
TELEFAX (051) 76.52.52